UNIVERSIDAD CARLOS III DE MADRID

PROYECTO FIN DE CARRERA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

FIRMA: APLICACIÓN DE FIRMA DIGITAL PARA FACTURAS ELECTRÓNICAS

MAYO 2010

Autor:

Esteban Gálvez Martín

Tutor:

José María Sierra Cámara

A todos los que quiero y me han apoyado

en esta misión que parecía nunca tener fin.

Agradecimientos

Una vez llegado a este punto, puedo decir que he alcanzado el fin de una etapa de mi vida, por eso desde estas líneas quiero expresar un profundo agradecimiento a quienes con su ayuda, fuerza y apoyo me han permitido alcanzar este sueño.

Mención especial a mis padres que con su esfuerzo y sacrificio me han convertido en la persona que soy y sin ellos esto no podía haber sido posible. En especial a ti, mamá, que seguro desde donde estés, lo estás viviendo tan emocionada como yo.

A mi hermano, Rubén, porque siempre está ahí y para que sepa que he terminado antes que él.

A al resto de mi familia porque siempre están tanto en los momentos buenos como en los malos.

A José María Sierra Cámara por el apoyo que me ha brindado durante toda mi etapa universitaria y en especial, durante la elaboración de este Proyecto.

Índice

Índice de ilustraciones

Índice de tablas

I. Introducción

1.1 Motivación

Una factura es el justificante fiscal de la entrega de un producto o de la provisión de un servicio, que afecta al obligado tributario emisor (el vendedor) y al obligado tributario receptor (el comprador). Tradicionalmente, es un documento en papel, cuyo original debe ser archivado por el receptor de la factura. Habitualmente el emisor de la factura conserva una copia o la matriz en la que se registra su emisión.

La factura electrónica es el equivalente digital y evolución lógica de la tradicional factura en papel. A diferencia de ésta, se emplean soportes informáticos para su almacenamiento en lugar de un soporte físico como es el papel.

En los países en los que la legislación lo admite, la validez de una factura electrónica es exactamente la misma que la de la tradicional factura en papel y gracias a la firma digital que incluye se garantiza su integridad y un alto nivel de trazabilidad, por lo que judicialmente es un documento considerado como vinculante y que no necesita de mayor prueba o confirmación que su propia existencia.

La factura electrónica es un tipo de factura que se diferencia de la factura en papel por la forma de gestión informática y el envío mediante un sistema de comunicaciones que conjuntamente permiten garantizar la autenticidad y la integridad del documento electrónico.

Una factura electrónica se construye fundamentalmente en un proceso que se puede dividir en dos grandes fases:

1. Se crea la factura tal y como se ha hecho siempre y se almacena en un fichero de datos.

2. Posteriormente se procede a su firma con un certificado digital o electrónico propiedad del emisor que cifra el contenido de factura y añade el sello digital a la misma.

Al terminar obtenemos una factura que nos garantiza:

- En primer lugar, que la persona física o jurídica que firmó la factura es quien dice ser (autenticidad) y
- En segundo lugar, que el contenido de la factura no ha sido alterado (integridad).

El emisor envía la factura al receptor mediante medios electrónicos, como pueden ser CDs, memorias Flash e incluso Internet. Si bien se dedican muchos esfuerzos para unificar los formatos de factura electrónica, actualmente está sometida a distintas normativas y tiene diferentes requisitos legales exigidos por las autoridades tributarias dependiendo de cada país, de forma que no siempre es posible el uso de la factura electrónica, especialmente en las relaciones con empresas extranjeras que tienen normativas distintas a la del país de origen.

Los requisitos legales respecto al contenido mercantil de las facturas electrónicas son exactamente las mismas que regulan las tradicionales facturas en papel. Los requisitos legales en relación con la forma imponen un determinado tratamiento con el fin de garantizar la integridad y la autenticidad y ciertos formatos que faciliten la interoperabilidad.

Existen algunas normativas internacionales aplicables de forma general a la factura electrónica, aunque las Naciones Unidas, a través de UN/CEFACT han publicado recomendaciones tales como UNeDocs, que definen plantillas para las facturas impresas y formatos EDI y XML para las modalidades electrónicas. En Europa, la facturación electrónica se regula en la Directiva 115/2001, que debía ser adoptada en cada país antes del 31 de diciembre de 2003.

Hoy día la organización GS1 (antes EAN/UCC) a nivel mundial ha organizado comités internacionales de usuarios de 108 países miembros, para conformar las guías de facturación electrónica estándar a nivel mundial.

La factura electrónica permite que instituciones, empresas y profesionales dejen atrás las facturas en papel y las reemplacen por la versión electrónica del documento tributario.

Tiene exactamente la misma validez y funcionalidad tributaria que la factura tradicional en papel. Todo el ciclo de la facturación puede ser administrado en forma electrónica.

Una vez realizado el estudio sobre los diferentes aspectos legales concernientes a la facturación electrónica en España, en el que se obtiene como resultado el estándar adoptado por las Administraciones Públicas es el formato Facturae obtenemos la principal motivación de la realización de este Proyecto. Partiendo de facturas electrónicas contenidas en ficheros PDF, dotar al usuario de una aplicación con funcionalidad de firma digital para dichas facturas y permitirle comprobar la finalidad de las mismas, como anteriormente se ha comentado:

- Por un lado, que la persona física o jurídica que firmó la factura es quien dice ser (autenticidad) y
- En segundo lugar, que el contenido de la factura no ha sido alterado con fecha posterior a su firma (integridad).

1.2 Resumen

La finalidad de este proyecto es dotar al usuario final de una aplicación que permita tanto firmar digitalmente facturas en formato electrónico PDF mediante certificados digitales almacenados en tarjetas criptográficas, como la verificación de las firmas digitales de las facturas.

Para ello se ha realizado una aplicación, llamada FIRMA, que implementada en modo Cliente-Servidor y apoyándose en el API de Facturae desarrollado por el Ministerio de Industria, Comercio y Turismo permite alcanzar esta finalidad.

1.3 Objetivos

Antes de abordar cualquier aspecto relacionado con este Proyecto, es necesario establecer una serie de objetivos o metas que se quieren alcanzar mediante la finalización del mismo.

A continuación, se enumeran los distintos objetivos que se marcaron inicialmente para la realización de este Proyecto:

- Dotar al cliente final de una aplicación capaz de firmar facturas electrónicas utilizando para ello el estándar empleando en las Administraciones Públicas españolas, el esquema Facturae.
- Permitir al usuario la verificación de la autenticación e integridad de una factura firmada con la aplicación.
- Facilitar al usuario un método útil y de fácil manejo para la transmisión de la factura y su firma a terceros. En este caso, se ha optado por dotar a la aplicación con la funcionalidad de ser capaz de enviar dichos ficheros por correo electrónico.

1.4 Estructura de la memoria

A lo largo de esta memoria se documenta estructuradamente el trabajo realizado durante el desarrollo de este Proyecto. El contenido de ésta se puede estructurar en las siguientes partes o bloques:

- Introducción.
- Antecedentes.
- Diseño de la aplicación.
- Implementación.
- Test y resultados.
- Futuros trabajos.
- Conclusiones.
- Bibliografía
- Anexos.

Introducción

Pretende acercar de forma superficial al lector a la idea general del Proyecto, así como al resumen del mismo y sus objetivos.

Antecedentes

En este apartado están descritas punto por punto todas las tecnologías, herramientas y metodologías empleadas para la realización de este Proyecto.

Diseño de la aplicación

En este punto se recoge el diseño de la aplicación apoyándose en varias herramientas y metodologías UML como son casos de uso, diagramas de actividad, etc.

Implementación

En este apartado se encuentran recogida toda aquella información importante referente a la implementación de la aplicación.

Test y resultados

Este apartado contiene las diferentes pruebas llevadas a cabo para comprobar el funcionamiento de la aplicación y los resultados que de ellas se han obtenido.

Futuros trabajos

En este punto se recogen los diferentes trabajos cuya elaboración implique de un cierto modo a la aplicación FIRMA con la posibilidad de ser desarrollados en un futuro próximo.

Conclusiones

En dicho punto se encuentra la conclusión y la opinión personal del autor referente a este Proyecto.

Bibliografía

En este apartado está toda una relación de toda la bibliografía empleada durante la elaboración de este Proyecto.

Anexos

En este último apartado, se encuentran los anexos incluidos a esta memoria. Se pueden encontrar los siguientes:

- Estudio sobre la planificación y presupuesto del proyecto.
- Manual de instalación de la aplicación FIRMA, tanto de la parte servidor (Servidor FIRMA) como la instalación en los clientes (Cliente FIRMA).
- Manual de usuario de la aplicación FIRMA.
- Glosario de términos.

II. Antecedentes

A continuación se enumerarán y expondrán las diferentes tecnologías y herramientas empleadas durante la realización de este Proyecto:

2.1 Java

Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++ pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, como la manipulación directa de punteros.

El lenguaje Java se creó con cinco características principales:

- Usa la metodología de la programación orientada a objetos.
- Permite la ejecución de un mismo programa en múltiples sistemas operativos.
- Incluye, por defecto, soporte para el trabajo en red.
- Está diseñado para ejecutar código en sistemas remotos de forma segura.
- Es fácil de usar y coge lo mejor de otros lenguajes orientados a objetos, como C++.

Orientado a objetos

La primera característica, orientado a objetos, se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para orientado a objetos, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos.

Un objeto puede verse como un paquete que contiene el comportamiento, el código, y el estado, los datos.

El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos.

Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas, objetos, que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico *cliente*, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de código abierto, quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.

Independencia de la plataforma

Significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez

y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, *"Write once, run everywhere"*, "Escríbelo una vez, ejecútalo en cualquier lugar".

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como *bytecode*, específicamente Java bytecode, es decir, instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual Java (JVM), un programa escrito en código nativo de la plataforma destino, que es el que entiende su hardware, que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo, como los gráficos, ejecución mediante threads, la interfaz de red, de forma unificada.

Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como GCJ (GNU Compiler for Java). Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

Las primeras implementaciones del lenguaje usaban una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase una reputación de lento en rendimiento.

Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas, aunque sigue siendo más lento que otros lenguajes.

La primera de estas técnicas es simplemente compilar directamente en código nativo como hacen los compiladores tradicionales, eliminando la etapa del bytecode. Esto da lugar a un gran rendimiento en la ejecución, pero tapa el camino a la portabilidad.

Otra técnica, conocida como compilación JIT (Just In Time, o *compilación al vuelo*), convierte el bytecode a código nativo cuando se ejecuta la aplicación. Otras máquinas virtuales más sofisticadas usan una *recompilación dinámica* en la que la máquina virtual es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado de optimización que la compilación tradicional o estática, ya que puede basar su trabajo en el conocimiento que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria. La compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad en ambos.

El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el entorno del servidor, como los Servicios Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.

El recolector de basura

En Java el problema de las fugas de memoria se evita en gran medida gracias a la recolección de basura (o *automatic garbage collector*). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el responsable de gestionar el ciclo de vida de los objetos. El programa, u otros objetos pueden tener localizado un objeto mediante una referencia a éste. Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas, como por ejemplo, un objeto creado y únicamente usado dentro de un método sólo tiene entidad dentro de éste; al salir del método el objeto es eliminado.

Aun así, es posible que se produzcan fugas de memoria si el código almacena referencias a objetos que ya no son necesarios, es decir, pueden aún ocurrir, pero en un nivel conceptual superior.

En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos, mayor seguridad y puede que más rápida que en C++.

Para finalizar indicar que Java puede ser utilizado para la realización de dos tipos de programas: aplicaciones independientes y los applets.

Las aplicaciones independientes son las que tienen el mismo comportamiento y funcionalidad independientemente del lenguaje empleado para su desarrollo.

Un applet, es un programa que puede mezclarse en un documento HTML, es decir en una página web. Cuando desde un navegador cargamos una página web que contiene un applet, éste se descarga y comienza a ejecutarse consiguiendo su objetivo primordial: proporcionar una forma fácil de ejecutar aplicaciones desde el navegador web.

Para obtener las últimas novedades sobre el API de Java o la última versión visitar la página web oficial: http://java.sun.com.

2.1.1 Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma. Fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse, actualmente es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Eclipse forma también una gran comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

El entorno de desarrollo integrado (IDE) de Eclipse emplea distintos módulos, *plug-ins,* para proporcionar distintas funcionalidades al usuario, a diferencia de otros entornos

monolíticos donde las funcionalidades están todas incluidas, se necesiten o no, esta arquitectura permite a Eclipse extenderse usando otros lenguajes de programación como, por ejemplo C, C++ y Python, trabajar con lenguajes para procesado de texto, aplicaciones en red como Telnet entre otras.

La definición que da el proyecto Eclipse acerca de su software es: "*Una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular*".

El sitio web oficial es el siguiente: http://www.eclipse.org, en el que podremos encontrar diversa documentación acerca del IDE y las últimas versiones.

2.1.2 ICEpdf

Debido a la importancia de este API utilizado en la implementación de este Proyecto, es necesario detallarlo y aportar alguna información relevante acerca de él.

ICEpdf es una librería Java de código abierto desarrollada por ICEsoft Technologies Inc., que permite visualizar ficheros PDF desde una aplicación Java sin problemas. Se puede utilizar de dos formas distintas:

- Como un servlet, en una aplicación web, que se encarga de cargar un PDF (ya sea que se haga un upload de un archivo local o se cargue alguno ya existente en la aplicación), lo transforma a una imagen y permite desplegarlo en el navegador.
- Embebido en aplicaciones Swing, como en el caso de FIRMA, como un componente que permite visualizar y manipular los ficheros PDF. Por manipular simplemente hay que entender a hacer operaciones típicas sobre este tipo de documentos: Buscar, Zoom, Imprimir, Guardar, etc.

La principal ventaja del uso de esta herramienta es que no se depende de un visor externo, como puede ser habitualmente, el Acrobat Reader, por ello, permite al programador tener el control total sobre las tareas a permitir y las tareas a impedir al usuario final con los documentos mostrados.

Las últimas novedades acerca de ICEpdf se pueden seguir en la página web oficial: http://www.icepdf.org.

2.2 Tarjetas inteligentes

Una tarjeta inteligente, *smart card*, o tarjeta con circuito integrado (TCI), es cualquier tarjeta del tamaño de un bolsillo con circuitos integrados que permiten la ejecución de cierta lógica programada. Aunque existe un diverso rango de aplicaciones, hay dos categorías principales de TCI. Las tarjetas de memoria, que contienen sólo componentes de memoria no volátil y posiblemente alguna lógica de seguridad, y las tarjetas microprocesadoras, que contienen memoria y microprocesadores.

La percepción estándar de una tarjeta inteligente es una tarjeta microprocesadora de las dimensiones de una tarjeta de crédito, también puede ser más pequeñas, como por ejemplo, las tarjetas SIM o GSM, con varias propiedades especiales, por ejemplo, un procesador criptográfico seguro, sistema de archivos seguro, características legibles por humanos, y es capaz de proveer servicios de seguridad como confidencialidad de la información en la memoria.

Las tarjetas no contienen baterías, la energía es suministrada por los lectores de tarjetas.

Ilustración 1: Muestra de tarjetas inteligentes

Existen numerosas clasificaciones de tarjetas inteligentes dependiendo de sus capacidades, de la estructura de su sistema operativo, de su tamaño, e interfaz. Por ello nos centraremos en su clasificación en función de sus capacidades como ejemplo.

En función de las capacidades que tiene el chip de una tarjeta inteligente se pueden catalogar los siguientes tipos:

- Memoria: tarjetas que tienen como fin ser un contenedor de ficheros pero que no albergan aplicaciones ejecutables. Se emplean para la identificación y control de acceso sin altos requisitos de seguridad.
- Microprocesadas: tarjetas con una estructura análoga a la de un ordenador, ya que disponen de procesador, memoria volátil y memoria permanente. Éstas si contienen ficheros y aplicaciones. Suelen emplearse para la identificación y pago como monederos electrónicos.
- Criptográficas: Son tarjetas microprocesadas avanzadas en las que hay módulos hardware que permiten la ejecución de algoritmos usados en cifrados y firmas digitales. En estas tarjetas se puede almacenar de forma segura uno o varios certificados digitales, y sus claves privadas, y firmar documentos o autenticarse

con la tarjeta sin que el certificado abandone la tarjeta, ya que es el procesador de la propia tarjeta el que realiza la firma o la autenticación. Como ejemplo de estas tarjetas, son las emitidas por la Fábrica Nacional de Moneda y Timbre (FNMT) y el Documento Nacional de Identidad electrónico (DNIe). Estos dos últimos ejemplos, serán los utilizados en la realización de este Proyecto.

2.2.1 Documento Nacional de Identidad electrónico (DNIe)

En España se expide desde marzo del año 2006 un tipo especial de documento de identidad denominado DNI electrónico.

El nacimiento del Documento Nacional de Identidad electrónico (DNIe) responde a la necesidad de otorgar identidad personal a la ciudadanía para su uso en la nueva Sociedad de la Información, además de servir de impulsor de la misma. Así, el DNIe es la adaptación del documento de identidad tradicional a la nueva realidad de una sociedad interconectada por las redes de comunicaciones. De este modo, cada ciudadano podrá hacer realizar múltiples gestiones de forma segura a través de medios telemáticos y asegurando la identidad de los participantes en la comunicación.

La adjudicataria del proyecto es la Unión Temporal de Empresas, compuesta por Indra, Telefónica y Software AG.

Ilustración 2: Tarjeta Criptográfica (DNIe)

En el DNI electrónico se han desarrollado tres niveles de seguridad. En un primer nivel, podemos encontrar tanto hologramas, letras táctiles como imágenes láser cambiantes. En un segundo nivel, imágenes codificadas, microtextos, kinegramas; y, por último, medidas criptográficas y biométricas.

El microchip, que constituye la principal novedad visible por el usuario, almacena la siguiente información:

- Los datos de filiación del titular, correspondientes con el contenido personalizado en la tarjeta.
- La fotografía digitalizada del ciudadano.
- La imagen digitalizada de la firma manuscrita.
- La plantilla biométrica de la impresión dactilar de los dedos índice de cada mano.
- Certificados.

Dentro de los certificados, podemos encontrar además del certificado de la Autoridad de Certificación expedidora y de un certificado de componente que permite la autenticación mutua de dispositivos tal y como se describe en el estándar CWA-14890, dos certificados X.509 del ciudadano, uno de autenticación y otro de firma, con la misma validez jurídica que la firma manuscrita, y las claves privadas asociadas a cada uno de ellos. Es conveniente precisar que cada pareja de claves se genera dentro del chip durante el proceso de expedición.

El hecho de que haya dos certificados persigue que el ciudadano pueda distinguir entre las actividades de autenticación y firma electrónica cuando se produzcan, al margen de la similitud de los procesos criptográficos implicados en ambas.

El DNI electrónico presenta una serie de ventajas para el ciudadano entre las que hay que destacar las siguientes:

- Desde el punto de vista de la **SEGURIDAD:**
    - El DNI electrónico es un documento más seguro que el tradicional, pues incorpora mayores y más sofisticadas medidas de seguridad que harán virtualmente imposible su falsificación.
    - Mediante el DNI electrónico podremos garantizar la identidad de los interlocutores de una comunicación telemática, ya sea para intercambio

de información, acceso a datos o acciones o compra por Internet. Igualmente, gestionar mejor el acceso a nuestro espacio de trabajo, nuestro ordenador personal y a la información que contenga.

- Usando el DNI electrónico podemos intercambiar mensajes con la certeza de que nuestro interlocutor es quien dice ser y que la información intercambiada no ha sido alterada.

- Desde el punto de vista de la **COMODIDAD**:
  - Con el DNI electrónico se podrán realizar trámites a distancia y en cualquier momento: El DNI electrónico permitirá realizar multitud de trámites sin tener que acudir a las oficinas de la Administración y sin tener que guardar colas. Y hacerlo en cualquier momento (24 horas al día, 7 días a la semana).
  - El DNI electrónico se expedirá de forma inmediata: No será necesario acudir dos veces a la Oficina de Expedición, sino que la solicitud y la obtención del documento se hará en una única comparecencia, en cualquiera de las Oficinas de Expedición existentes en España, que se irán dotando progresivamente del equipamiento necesario para la expedición del nuevo documento.
  - Hacer trámites sin tener que aportar una documentación que ya exista en la Administración: una de las ventajas derivadas del uso del DNI electrónico y de los servicios de Administración Electrónica basados en él será la práctica eliminación del papel en la tramitación. El ciudadano no tendrá que aportar una información que ya exista en otra Unidad de la Administración, evitándose -de nuevo- colas y pérdidas de tiempo. La Unidad que realice la tramitación lo hará por él, siempre que el ciudadano así lo autorice.
- Desde el punto de vista de la **ERGONOMÍA**:
  - El DNI electrónico es un documento más robusto. Está construido en policarbonato y tiene una duración prevista de unos diez años.
  - El DNI electrónico mantiene las medidas del DNI tradicional (idénticas a las tarjetas de crédito habituales).

A continuación se expone una pequeña descripción sobre el DNIe adjuntando una ilustración para que sea más fácil su identificación.

En el anverso de la tarjeta se encuentran los siguientes elementos:

- En el cuerpo central:
  - PRIMER APELLIDO: Primer apellido del ciudadano.
  - SEGUNDO APELLIDO: Segundo apellido del ciudadano.
  - NOMBRE: Nombre del ciudadano.
  - SEXO Y NACIONALIDAD: Sexo y nacionalidad del ciudadano.
  - FECHA DE NACIMIENTO: Fecha de nacimiento del ciudadano.
  - IDESP: Número de serie del soporte físico de la tarjeta.
  - VÁLIDO HASTA: Fecha de validez del documento.
- En la esquina inferior izquierda:
  - DNI NUM: Número del Documento Nacional de Identidad del ciudadano, seguido del carácter de verificación (Número de Identificación Fiscal).
- En el espacio destinado a la impresión de imagen laser cambiante (CLI):
  - La fecha de expedición en formato DDMMAA.
  - La primera consonante del primer apellido + primera consonante del segundo apellido + primera consonante del nombre (del primer nombre en caso de ser compuesto).
- Chip criptográfico, cuyo contenido ya se ha indicado anteriormente.
- Elementos de seguridad del documento para impedir su falsificación:
  - Medidas de seguridad físicas:
    - Visibles a simple vista (tintas ópticamente variables, relieves, fondos de seguridad).
    - Verificables mediante medios ópticos y electrónicos (tintas visibles con luz ultravioleta, microescrituras).
  - Medidas de seguridad digitales:
    - Encriptación de los datos del chip.
    - Acceso a la funcionalidad del DNI electrónico mediante clave personal de acceso (PIN).
    - Las claves nunca abandonan el chip.
    - La Autoridad de Certificación es la Dirección General de la Policía.

El reverso de la tarjeta contiene los siguientes elementos:

- Información impresa (y visible a simple vista) sobre la identidad del ciudadano en la parte superior:
  - LUGAR DE NACIMIENTO.
  - PROVINCIA-PAÍS.
  - HIJO DE.
  - DOMICILIO.
  - LUGAR DE DOMICILIO.
  - PROVINCIA-PAÍS Y EQUIPO.
- Información impresa OCR-B para lectura mecanizada sobre la identidad del ciudadano según normativa OACI para documentos de viaje.

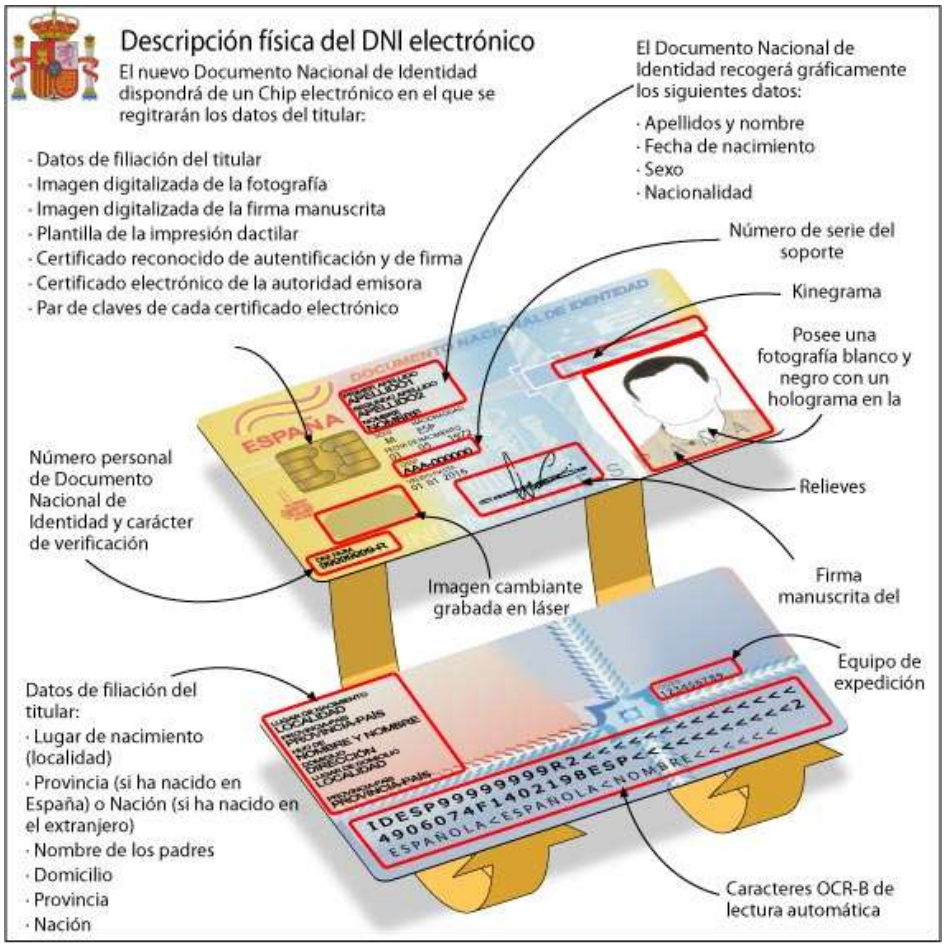

Ilustración 3: Descripción del DNIe ***(fuente: portal oficial www.dnielectronico.es)***

El uso del nuevo DNI electrónico requiere que el usuario recuerde la clave que se le asignó cuando lo obtuvo (PIN) y que puede cambiar en sistemas automatizados instalados en las dependencias policiales en las que se expide el DNI. Para ello solo es necesario identificarse con la huella dactilar.

Para la correcta utilización por parte del ciudadano del DNI electrónico en trámites telemáticos es necesario contar con determinados elementos hardware y software que nos van a permitir el acceso al chip de la tarjeta y, por tanto, la utilización de los certificados contenidos en él.

- Elementos hardware
  - Un ordenador personal (Intel –a partir de Pentium III- o tecnología similar).
  - Un lector de tarjetas inteligentes que cumpla las siguientes características:
    - Cumpla el estándar ISO-7816 (1, 2 y 3).
    - Soporte tarjetas asíncronas basadas en protocolos T=0 (y T=1).
    - Soporte velocidades de comunicación mínimas de 9.600 bps.
    - Soporte los estándares:
      - API PC/SC (Personal Computes/Smart Card).
      - CSP (Cryptographic Service Provider, Microsoft).
    - API PKCS#11.

    Existen distintas implementaciones, bien integrados en el teclado, bien externos (conectados vía USB), bien internos o bien a través de una interfaz PCMCIA.

Ilustración 4: Tipos de lectores de tarjetas inteligentes

- Elementos software:
  - Sistemas operativos: El DNI electrónico puede operar en diversos entornos:
    - Microsoft Windows.
    - Linux.
    - Unix.
    - Mac.
  - Navegadores web: Es compatible con los siguientes navegadores:
    - Microsoft Internet Explorer (versión 6.0 o superior).
    - Mozilla Firefox (versión 1.5 o superior).
    - Netscape (versión 4.78 o superior).
  - Controladores / Módulos criptográficos: Para poder interaccionar adecuadamente con las tarjetas criptográficas en general y con el DNI electrónico en particular, el equipo ha de tener instalados unas aplicaciones software denominadas módulos criptográficos, que dependen del sistema operativo:
    - En un entorno Microsoft Windows, el equipo debe tener instalado un servicio que se denomina "Cryptographic Service Provider" (CSP).
    - En los entornos UNIX / Linux o Mac, podemos utilizar el DNI electrónico a través de un módulo criptográfico denominado PKCS#11.

    Tanto el CSP como PCKS#11 específico para el DNI electrónico pueden obtenerse en la dirección www.dnielectronico.es/descargas

2.2.2 Tarjeta criptográfica FNTM-RCM

La tarjeta criptográfica FNMT-RCM, premiada como el "Avance tecnológico más relevante de 2003" (Premio Golden Card) en el Congreso Internacional de Tarjetas, proporciona todas las funciones y características de seguridad necesarias para implementar un sistema de autentificación de usuario y soporte de confidencialidad. Resulta una tarjeta ideal

en servicios de securización de intercambio de mensajes o documentos, en aplicaciones de comercio electrónico o sistemas de certificación.

Es el soporte físico empleado por el proyecto CERES, que posibilita el uso del certificado de la FNMT-RCM para los servicios de la Administración Electrónica que los numerosos Organismos oficiales están ofreciendo al ciudadano: Ministerios de Economía y Hacienda, Justicia, Administraciones Públicas, Defensa, Centro Nacional de Inteligencia, Guardia Civil, Consejo General del Notariado, Comisión Nacional de la Energía, Oficina Española de Patentes y Marcas, así como diversas Comunidades Autónomas, Ayuntamientos, Diputaciones, Colegios Profesionales, etc.

Esta tarjeta multi-aplicación constituye el lugar idóneo para almacenar el material criptográfico asociado al usuario. Soporta las técnicas criptográficas más avanzadas, como es el caso del algoritmo de cifrado simétrico Triple-DES, el algoritmo de cifrado asimétrico RSA con manejo de claves de 1.024 bits, y la generación de funciones unidireccionales hash mediante el algoritmo SHA-1.

Ilustración 5: Tarjeta criptográfica FNMT-RCM

A continuación se muestran tanto las especificaciones técnicas como las características principales de esta tarjeta criptográfica:

CPU	SLE66CX320P, ST19XL34
ROM	32 Kbytes con SLE66CX320P y 96 Kbytes con ST19XL34
RAM	256 bytes (+ 700 bytes de RAM interna) (+ 1 Kbyte de XRAM) 4 Kbytes con ST19XL34
EEPROM	32 Kbytes (SLE66CX320P) 34 Kbytes (ST19XL34)
Operaciones Criptográficas	Permite el almacenamiento y uso de claves de 1024 bits y con el componente ST19XL34 de hasta 2.176 bits. Generación y verificación de firmas digitales RSA. Cifrado y descifrado RSA. Generación de claves RSA. Cifrado y descifrado Triple DES. Cifrado hash SHA-1.
Seguridad	Cifrado dinámico de memoria/buses con diferentes claves. Sensores para control de tensión y frecuencia. Generador real de números aleatorios. Módulo de cálculo de CRCs.
Aplicaciones	Almacenamiento y gestión de certificados digitales X.509v3. Correo seguro: cifrado y/o firma, descifrado y verificación. Conexión segura cliente-servidor. Windows logon.

Tabla 1: Ficha técnica tarjeta criptográfica FNMT-RCM

Características de seguridad	Autenticación interna Tarjeta-Terminal. Autenticación externa de usuario y de aplicación. Validación de PIN de usuario. Servicios de integridad mediante la generación y verificación de firmas digitales RSA. Generación de claves RSA en tarjeta. Mecanismos de confidencialidad para el intercambio seguro de claves de cifrado. "Zona de espejo" para evitar pérdida de datos si la tarjeta es extraída durante una operación.
Normalización	SO 7816-1 /-2 /-3 (T=0) y 7816-4 en estructura de ficheros y órdenes. Especificaciones de Interoperabilidad entre Ordenadores Personales y Tarjetas Inteligentes PC/SC. Interfaz criptográfico PKCS#11 versión 2.01 desarrollado para esta tarjeta. Personalización de la estructura de ficheros conforme al estándar PKCS#15. Interfaz criptográfico CSP para Microsoft desarrollado para esta tarjeta. Certificado de seguridad Common Criteria EAL4+

Tabla 2: Características tarjeta criptográfica FNMT-RCM

2.3 Cryptokit

La Fábrica Nacional de Moneda y Timbre-Real Casa de la Moneda (FNTM-RCM) ha lanzado junto con la empresa fabricante de lectores para tarjetas inteligentes C3PO un sistema de cifrado e identificación digital, llamado Cryptokit.

Se puede definir brevemente el producto Cryptokit como una herramienta de seguridad de tipo PKI (Public Key Infrastructure = Infraestructura de Clave Pública) con soporte en tarjeta criptográfica. El producto nace de la unión de dos elementos hardware principales: la tarjeta criptográfica de la FNMT-RCM y el lector de tarjetas inteligentes (LTC31) de C3PO, que junto con el software adecuado lo convierten en una de las herramientas más seguras para el control de acceso y la identificación de usuarios, ya sea a través de Internet/Intranet o a través de redes de área local (LAN).

Hay que precisar que como parte software incluida en el producto, nos encontramos como es habitual, con los manuales de usuario e instalación tanto de la tarjeta criptográfica como del lector, los drivers del lector, los módulos criptográficos para la utilización de la tarjeta tanto en aplicaciones Microsoft (CSP) como en Mozilla/Netscape (PKCS#11), una aplicación que nos permite firmar y cifrar documentos directamente desde el sistema operativo (Crypto-tool) y una serie de aplicaciones desarrolladas por la FNMT-RCM que nos permiten desde la carga de certificados a la tarjeta criptográfica hasta el desbloqueo de la tarjeta y cambio de PIN pasando por la elección del certificado por defecto de la tarjeta.

La creciente utilización de los medios informáticos y telemáticos en las relaciones administrativas, empresariales y comerciales se traduce en la demanda de los usuarios de servicios y mecanismos de seguridad con los que fortalecer la confianza en las transacciones electrónicas. Además de eficacia y economía, se exige que dichos medios ayuden a preservar los derechos y obligaciones de los usuarios y consumidores, en condiciones equivalentes a las que proporciona el trámite convencional, sobre todo cuando surgen discrepancias entre las partes o se persigue el fraude.

Este sistema hace posible la seguridad de los datos, mediante el cifrado/descifrado de cualquier fichero utilizando algoritmos de alta seguridad; la seguridad en las comunicaciones,

envío de correo firmado y/o cifrado de forma segura, identificación segura a través de la red y es compatible con iniciativas de la Administración Pública incluidas en el Plan Info XXI que requieran el uso de Firma Digital y/o autentificación en su operativa a través de Internet. La tarjeta de identificación electrónica es un importante elemento de seguridad por su posible aplicación a numerosas gestiones, presentes y futuras, de la Administración Pública.

Ilustración 6: Cryptokit

La tarjeta criptográfica es la opción más segura para evitar el uso fraudulento de la propia identidad y el acceso a datos confidenciales. Tiene capacidad para generar y almacenar varios pares de claves, almacenar distintos certificados (Certificados de la FNMT-RCM, VeriSign, Entrust, etc.), datos adicionales del usuario; cifrado, firma y verificación en la propia tarjeta y securización de su uso mediante un código secreto.

Para más especificaciones y características sobre la tarjeta criptográfica se puede revisar en esta misma memoria el apartado *2.2.2 Tarjeta criptográfica FNMT-RCM.*

2.4 Factura electrónica

2.4.1 Definición

La facturación electrónica es un equivalente funcional de la factura en papel y consiste en la transmisión de las facturas o documentos análogos entre emisor y receptor por medios

electrónicos (ficheros informáticos) y telemáticos (de un ordenador a otro), firmados digitalmente con certificados reconocidos.

De esta definición extendida en todo el mercado, se transmite tres condicionantes para la realización de la facturación electrónica:

- Se necesita un formato electrónico de factura de mayor o menor complejidad (EDIFACT, XML, PDF, HTML, DOC, XLS, GIF, JPEG o TXT, entre otros).
- Es necesario una transmisión telemática (tiene que partir de un ordenador, y ser recogida por otro ordenador).
- Este formato electrónico y transmisión telemática, deben garantizar su integridad y autenticidad a través de una firma electrónica reconocida.

El artículo 3.3 de la Ley 59/2003 de 19 de diciembre define la firma electrónica reconocida como:

"la firma electrónica avanzada basada en un certificado reconocido y generada mediante un dispositivo seguro de creación de firma".

Es decir, se tienen que dar tres condicionantes para que se dé la firma electrónica reconocida:

- Que sea una firma electrónica avanzada, es decir:

"aquella que permite identificar al firmante y detectar cualquier cambio ulterior de los datos firmados, que está vinculada al firmante de manera única y a los datos a que se refiere y que ha sido creada por medios que el firmante puede mantener bajo su exclusivo control" (Art 2 de la misma ley).

- Que esté basada en un certificado reconocido, siendo certificado reconocido aquél que:

"cumpla los requisitos establecidos en esta Ley en cuanto a la comprobación de la identidad y demás circunstancias de los solicitantes".

- Que sea generada mediante un dispositivo seguro de creación de firma, es decir, aquel que ofrece, al menos, las siguientes garantías:

- Que los datos utilizados para la generación de firma pueden producirse sólo una vez y asegura razonablemente su secreto.
- Que existe una seguridad razonable de que los datos utilizados para la generación de firma no pueden ser derivados de los de verificación de firma o de la propia firma y de que la firma está protegida contra la falsificación con la tecnología existente encada momento.
- Que los datos de creación de firma pueden ser protegidos de forma fiable por el firmante contra su utilización por terceros.
- Que el dispositivo utilizado no altere los datos o el documento que deba firmarse ni impide que éste se muestre al firmante antes del proceso de firma. (Art. 24.3).

El Anteproyecto de Ley de Medidas de Impulso de la Sociedad de la Información define la factura electrónica como:

"un documento electrónico que cumple con los requisitos legal y reglamentariamente exigibles a las facturas y que, además, garantiza la autenticidad de su origen y la integridad de su contenido, lo que impide el repudio de la factura por su emisor".

La información sobre los prestadores que emiten certificados reconocidos se puede encontrar en dos fuentes:

- El Ministerio de Industria, Comercio y Turismo en virtud del artículo 30 de la Ley 59/2003, de 19 de diciembre, de firma electrónica, publica la información referente a los prestadores de servicios de certificación admitidos en su registro. A 16 de febrero de 2010, los prestadores que prestan servicios de certificación basados en certificados reconocidos que recoge dicho registro son:
  - Asociación Nacional de Fabricantes (ANF-AC).
  - Fábrica Nacional de Moneda y Timbre–Real Casa de la Moneda (CERES).
  - Certificados Camerales (CAMERFIRMA).
  - FIRMAPROFESIONAL.
  - Autoridad de Certificación de la Comunidad Valenciana (ACCV).
  - Consejo General de la Abogacía Española (AC ABOGACÍA).
  - Colegio de Ingenieros de Caminos, Canales y Puertos (CICCP).
  - Agencia Notarial de Certificación (ANCERT).

- BANESTO.
- IZENPE.
- Agencia Catalana de Certificación (CATCERT).
- IpsCA.
- Telefónica Empresas.
- Servicio de Certificación de los Registradores (SCR).
- Dirección General de la Policía (DGP).
- Colegio Oficial de Arquitectos de Sevilla (COAS).
- Banco Santander Central Hispano (BSCH).
- CertiVeR.
- Negonation (Tractis).
- Edicom.
- Ministerio de Defensa.
- Healthsign.
- Servicio de Salud de Castilla La Mancha.
- Organización Médica Colegial (OMC).
- European Agency of Digital Trust (EAD TRUST).
- Tesorería General de la Seguridad Social - GISS (Gerencia de Informática de la Seguridad Social).

- La AEAT publica en su web todos los certificados que a día de hoy están reconocidos a tal efecto y que quedan recogidos en las siguientes tablas obtenidas de la propia web de la Agencia (www.aeat.es).

Organización (O)	Unidad Organizativa (OU)	Nombre Común (CN)
AC Camerfirma S.A.		RACER
AC Camerfirma S.A.		AC Camerfirma Certificados
Agencia Catalana de Certificación	Secretaría de Administración y Funciones Públicas	EC-SAFP Sólo certificados de clase 1
Agencia Catalana de Certificación	Administraciones Locales de Cataluña	EC-AL Sólo certificados de clase 1
Agencia Catalana de Certificación	Entidad pública de certificación de ciudadanos	EC-IDCat
Agencia Catalana de Certificación	Universidades e Investigación	EC-UR
Agencia Notarial de Certificación S.L.		ANCERT Certificados Notariales Personales

Agencia Notarial de Certificación S.L.		ANCERT Certificados para empleados
Agencia Notarial de Certificación S.L.		ANCERT Certificados FERN
Agencia Notarial de Certificación S.L.		ANCERT Certificados para Corporaciones de Derecho Público
ANF Autoridad de Certificación	ANF Clase 1 CA	ANF Server CA
Banco Santander Central Hispano S.A.	Dirección de Universidades	WG10 Qualified Identification Root CA
Banesto S.A		PSC Banesto Clientes
Consejo General de la Abogacía	Consulte http://www.acabogacia.org	Autoridad de Certificación de la Abogacía
Consejo Superior de Cámaras de Comercio, Industria y Navegación		Consejo Superior de Cámaras
DIRECCIÓN GENERAL DE LA POLICIA	DNIE	AC DNIE 001
DIRECCIÓN GENERAL DE LA POLICIA	DNIE	AC DNIE 002
DIRECCIÓN GENERAL DE LA POLICIA	DNIE	AC DNIE 003
Firma profesional S.A.	Jerarquía de Certificación Firma profesional	AC Firma profesional – CA1
FNMT	FNMT Clase 2 CA	
Generalitat Valenciana	PKIGVA	PKIGVA
Generalitat Valenciana	PKIGVA	CAGVA
IZENPE S.A.	Certificado público SCI	CA de Ciudadanos y Entidades
Servicio de Certificación del Colegio de Registradores (SCR)	Certificado Raíz Certificado Propio	Certificado de Clave Secundaria para certificados externos
Servicio de Certificación del Colegio de Registradores (SCR)	Certificado Raíz Certificado Propio	Certificado de Clave Secundaria para certificados internos

Tabla 3: Listado de certificados válidos para personas físicas

Organización (O)	Unidad Organizativa (OU)	Nombre Común (CN)
AC Camerfirma S.A.		RACER
AC Camerfirma S.A.		AC Camerfirma Certificados Camerales
Agencia Catalana de Certificación	Administraciones Locales de Cataluña	EC-AL Sólo certificados de clase
Agencia Catalana de Certificación	Universidades e Investigación	EC-UR Sólo certificados de clase 1

Agencia Catalana de Certificación	Secretaría de Administración y Funciones Públicas	EC-SAFP Sólo certificados de clase 1
Agencia Notarial de Certificación S.L.		ANCERT Certificados Notariales Corporativos
ANF Autoridad de Certificación	ANF Clase 1 CA	ANF Server CA
Banesto S.A.		PSC Banesto Clientes
Consejo General de la Abogacía	Autoridad de Certificación de la Abogacía	ACA–Certificados Corporativos
Consejo Superior de Cámaras de Comercio, Industria y Navegación		Consejo Superior de Cámaras
Firmaprofesional S.A.	Jerarquía de Certificación Firmaprofesional	AC Firmaprofesional – CA1
FNMT	FNMT Clase 2 CA	
Generalitat Valenciana	PKIGVA	ACCV-CA1
IZENPE S.A.	Certificado público SCI	CA de Ciudadanos y Entidades
Servicio de Certificación del Colegio de Registradores (SCR)	Certificado Raíz Certificado Propio	Certificado de Clave Secundaria para certificados externos

Tabla 4: Listado de certificados válidos para personas jurídicas

Por último y para que tuviera la facturación electrónica la misma validez legal que una factura en papel, se necesita el consentimiento de ambas partes (emisor y receptor).

Adicionalmente, y como requisito de todas las facturas independientemente de cómo se transmitan, en papel o en formato electrónico, el artículo 6 del RD 1496/2003 que regula el contenido de una factura establece que los campos obligatorios de una factura son:

- Número de factura.
- Fecha de expedición.
- Razón social emisor y receptor
- NIF emisor y receptor.
- Domicilio emisor y receptor.
- Descripción de las operaciones (base imponible).
- Tipo impositivo.
- Cuota tributaria.
- Fecha prestación del servicio (si es distinta a expedición).

En definitiva, para cumplir con la norma y que una factura electrónica tenga la misma validez legal que una emitida en papel, el documento electrónico que la representa debe contener los campos obligatorios exigibles a toda factura, estar firmado mediante una firma electrónica avanzada basado en certificado reconocido y ser transmitido de un ordenador a otro recogiendo el consentimiento de ambas partes.

2.4.2 Formato Facturae, estándar para las facturas electrónicas a nivel nacional

En España, el CCI (Centro de Cooperación Interbancaria), en común acuerdo con la Agencia Tributaria (AEAT), ha desarrollado un conjunto de recomendaciones para codificar tanto la factura electrónica como la firma electrónica. El formato está basado en XML y recoge todos los requisitos de la normativa española, en particular el RD-1496/2003 y el RD-87/2005. Surge entonces el formato para la factura electrónica denominado AEAT-CCI, desarrollado en las versiones 1.1, 1.2 y 2.0.

En base a este formato, a iniciativa de la Agencia Tributaria y del Ministerio de Industria, Turismo y Comercio, y de la necesidad de establecer un modelo válido en el que basarse la facturación electrónica, se publica el formato *Facturae,* cuya versión inicial es la 3.0 obtenida a partir del esquema AEAT-CCI versión 2.0, que dispone de un portal en http://www.facturae.es y se oficializa por la ORDEN PRE/2971/2007, de 5 de octubre, sobre la expedición de facturas por medios electrónicos cuando el destinatario de las mismas sea la Administración General del Estado u organismos públicos vinculados o dependientes de aquélla y sobre la presentación ante la Administración General del Estado o sus organismos públicos vinculados o dependientes de facturas expedidas entre particulares. (BOE n. 247 de 15/10/2007).

Este formato será de obligado cumplimiento en España para facturas remitidas a las Administraciones Públicas y evolucionará de forma que pueda ser considerado la personalización para España de los estándares internacionales y su guía de implantación.

A nivel general se puede dividir el esquema Facturae en cinco grandes bloques:

- **Primer bloque, *FileHeader*:** Se registran los datos generales. A resaltar que es en este bloque donde se indica si existe subfacturación y se identifica a este tercero. También se marca el lote de la remesa y si hay cesión de factoring de estos datos.
- **Segundo bloque, *Parties*:** Se identifica adecuadamente tanto al emisor como al receptor de la factura.
- **Tercer bloque, *Invoices*:** Se determinan los datos comunes de las facturas. Identificación de las facturas, bases imponibles fechas importes, etc.
- **Cuarto bloque, *Extensions*:** Extensión de las facturas, donde se determinan las reglas específicas de cada sector y de cada relación entre compañías. Este bloque es opcional.
- **Quinto bloque, *dsSignature*:** Recoge los datos de la firma. En el caso de que una factura no esté firmada, como es obvio, no se hallará este bloque.

En la ilustración siguiente se puede apreciar cada uno de estos bloques con la información principal que cada uno de ellos contiene.

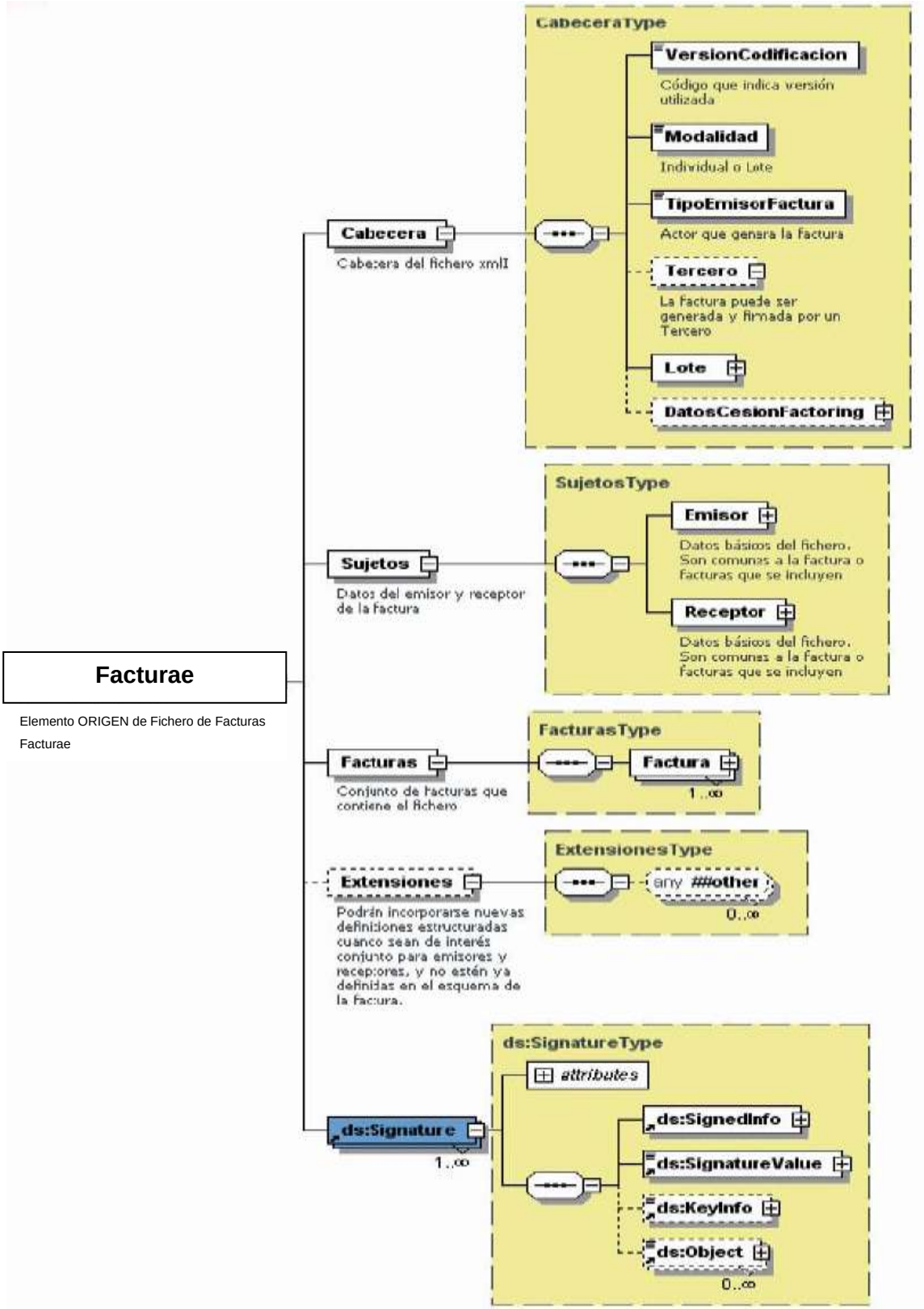

Ilustración 7: Esquema Facturae

Firma XML (dsSignature)

Debido a su importancia dentro del esquema Facturae se ha considerado necesario su explicación en un apartado aparte.

La utilización de firmas en XML es la elección evidente cuando el formato empleado para la formación de las propias facturas es también XML. Sin embargo, la firma XML puede aplicarse a cualquier tipo de documento, con independencia de su formato. Dentro de la firma electrónica en formato XML, existen diferentes "subtipos de formatos", dentro de los cuales destacan por encima de todos el XML DSig y la variante de éste, el XML Advanced Electronic Signatures (XAdES).

En toda firma XML, según el estándar XML DSig, existirían tres modos de firma:

- ***Enveloped***, en el que la firma se añade al final del documento XML como un elemento más. Se firma todo lo inmediatamente anterior al documento. Es el tipo empleado en el esquema Facturae.
- ***Enveloping***, en el que el documento se incluye dentro de la firma en la que se referencia lo firmado como objeto insertado en la firma. Ya que se referencian los objetos, este modelo permitiría distinguir lo que se firma, pudiendo firmar el objeto entero o partes de él (asignando un id diferenciador).
- ***Detached***, en el que la firma y el documento se separan en dos archivos, la URL donde se encuentra el documento puede aparecer en la propia firma.

XAdES, está basado en XML DSig, pero con la ventaja de añadir diversas capas de seguridad a la firma y al documento firmado. En concreto, se pueden agregar los datos relativos a la revocación y al sello de tiempo, tal y como se muestra siguiente en la siguiente ilustración:

```
XMLDSIG

<ds:Signature ID?>
    <ds:SignedInfo>
        <ds:CanonicalizationMethod/>
        <ds:SignatureMethod/>
        (<ds:Reference URI? >
            (<ds:Transforms>)?
            <ds:DigestMethod>
            <ds:DigestValue>
        </ds:Reference>)+
    </ds:SignedInfo>
```

```
    <ds:SignatureValue>
        (<ds:KeyInfo>)?
    </ds:SignatureValue>

    <ds:Object>
        <QualifyingProperties>
            <SignedProperties>
                <SignedSignatureProperties>
                    (SigningTime)
                    (SigningCertificate)
                    (SignaturePolicyIdentifier)
                    (SignatureProductionPlace)?
                    (SignerRole)?
                </SignedSignatureProperties>

                <SignedDataObjectProperties>
                    (DataObjectFormat)*
                    (CommitmentTypeIndication)*
                    (AllDataObjectsTimeStamp)*
                    (IndividualDataObjectsTimeStamp)*
                </SignedDataObjectProperties>

            </SignedProperties>

            <UnsignedProperties>
                <UnsignedSignatureProperties>
                    (CounterSignature)*
                </UnsignedSignatureProperties>
            </UnsignedProperties>
        </QualifyingProperties>
    </ds:Object>

</ds:Signature>

XADES
```

Ilustración 8: Esquema de agregación de datos relativos a la revocación y sello del tiempo

A pesar de que los datos relativos tanto al sellado de tiempo como a la revocación del certificado pueden ser de gran utilidad y aportar un gran valor añadido al proceso de firma, éstos no son obligatorios para dotar de validez a una factura electrónica.

XAdES cuenta con el respaldo de W3C, OASIS y ETSI (European Telecommunications Standards Institute) y está especificado en el estándar TS 101 903 v.1.2.2.

2.4.3 API Java para factura electrónica (Facturae)

Con objeto de facilitar la incorporación del formato Facturae en las aplicaciones de facturación electrónica ya existentes o en desarrollo, el Ministerio de Industria, Turismo y

Comercio, ha puesto en disposición de los usuarios interesados un API (Application Programming Interface) desarrollado en Java que implementa la funcionalidad necesaria para gestionar facturas con formato Facturae (versiones 3.0 y 3.1).

Se han desarrollado tres distribuciones del API dependiendo del navegador y/o sistema operativo del usuario, son las siguientes:

- API para Internet Explorer en Microsoft Windows.
- API para Mozilla en Microsoft Windows.
- API para Mozilla en Linux.

Cada una de estas distribuciones está empaquetada en un fichero ZIP que contiene:

- FacturaeAPI.jar: Fichero .jar del API.
- Directorio "lib": Contiene las librerías de apoyo, entre las que se encuentran los componentes de firma también desarrollados por el Ministerio.
- Directorio "META-INF": Incorpora el manifest.mf, disclaimer.txt (castellano e inglés), LEEME.txt (castellano e inglés), licencia.txt (castellano, inglés, catalán, gallego y euskera).
- Guía de uso del API (en formato PDF).

Para la realización de este Proyecto se ha empleado la distribución del API para Mozilla en Microsoft Windows, decisión tomada gracias a la seguridad y fiabilidad que nos otorga el navegador de Mozilla, y nos hemos decantando por el sistema Microsoft Windows ya que es uno de los más implantados.

Para poder entender, un poco por encima, la funcionalidad del API de Facturae se va a detallar a continuación su arquitectura.

Este API se divide en dos partes bien diferenciadas:

- Por un lado están los componentes de marshal, unmarshal y validación. Ayudan a la gestión de datos XML, permitiendo la conversión recíproca de datos XML y objetos Java.
- Por otro lado se encuentra el componente de firma, encargado de realizar la firma digital de una factura electrónica.

El API está compuesto por las siguientes librerías (cada grupo de librerías corresponde, respectivamente, a las partes diferenciadas comentadas anteriormente):

- Facturae-API.jar → Es el núcleo del API y contiene los componentes básicos.
- Librerías JAXB → Contiene la lógica necesaria para realizar la transformación entre datos XML y objetos Java. Permite realizar los procesos de marshal (Java →XML) y unmarshal (XML →Java).
    - activation.jar
    - jaxb-api.jar
    - jaxb-impl.jar
    - jsr173_1.0_api.jar
- swing-layout-1.0.3.jar → Contiene extensiones de los "layout" de swing. Son empleados por la interfaz de usuario de selección de certificado.
- Componentes de logs → Se utiliza la librería commons-logging.jar para flexibilizar el sistema de logs empleado por el desarrollador. Se incluye log4j por defecto.
    - commons-logging.jar
    - o log4j-1.2.15.jar
- FacturaEBridge.jar → "Bridge" de firma. Contiene la interfaz "fachada" de los servicios de firma.
- Facturae-API-CompPack.jar → Contiene el componente de firma empleado por el API. Se conecta al "bridge" anterior mediante configuración.
- LibXADESJNI_MZ_W5.jar → Librerías nativas de acceso al almacén de certificados de Mozilla en Windows.

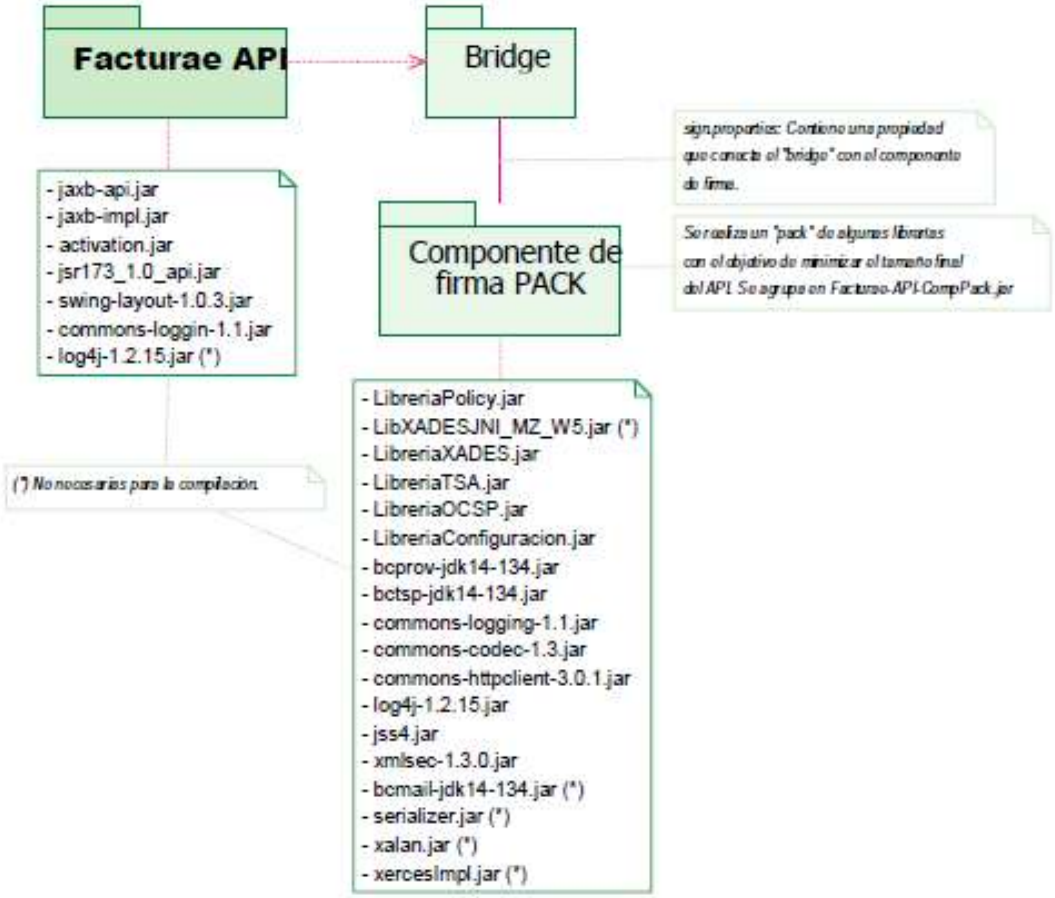

Ilustración 9: Arquitectura general del API Facturae *(fuente: Guía de usuario para el API de Facturae)*

III. Diseño

3.1 Descripción del sistema

El sistema consiste en una aplicación Cliente-Servidor desarrollada en Java que a su vez hace uso del API Facturae desarrollado por el Ministerio de Industria, Comercio y Turismo para el tratamiento de facturas en formato electrónico.

En nuestro caso, el formato de la factura electrónica será el indicado en el apartado *3.4 Diseño factura PDF* contenido en esta memoria. Para ello se utilizará un fichero PDF que contendrá a la factura siguiendo el anterior formato.

Gracias al uso de una arquitectura Cliente-Servidor, se dota al sistema de un carácter distribuido. Los clientes solicitarán al servidor las correspondientes peticiones que éste procesará. Estas peticiones se pueden englobar en dos tipos: en primer lugar de transformación y por último de verificación.

Las peticiones de transformación son aquellas en las que los clientes necesitan realizar una firma de una factura contenida en un fichero PDF, y para ello, envían dicho fichero al servidor para que éste realice la transformación de este formato al formato Facturae versión 3.1. Una vez procesada, el servidor devolverá al cliente correspondiente un fichero en formato Facturae para que proceda a su firma.

Las peticiones de verificación son las que se producen cuando los clientes necesitan verificar la integridad y autenticación de la firma de una factura.

La realización de la firma se lleva a cabo mediante el empleo del certificado seleccionado contenido en la tarjeta inteligente, ya sea DNIe, tarjeta criptográfica de la FNMT-RCM, en el Cliente FIRMA.

3.2 Arquitectura del sistema: Cliente-Servidor

La arquitectura elegida para la implementación de la aplicación FIRMA es la correspondiente a Cliente-Servidor.

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Como ejemplos específicos de servidores se incluyen los servidores web, los servidores de archivos, los servidores del correo, etc. Aunque sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

El Cliente FIRMA es remitente de las solicitudes y presenta las siguientes características:

- Es quien inicia las solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo maestro).
- Espera y recibe las respuestas del Servidor FIRMA, ya sean notificaciones o los ficheros de firma XSIG.
- Interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

El Servidor FIRMA es el receptor de las solicitudes enviadas por los Clientes FIRMA y presenta las siguientes características:

- Al iniciarse espera a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo esclavo).

- Tras la recepción de una solicitud, la procesa, es decir, realiza la transformación de la factura PDF a Facturae versión 3.1 (XSIG) o verifica la firma XSIG y luego envía la respuesta al cliente (fichero o notificación).
- Acepta conexiones desde un gran número de clientes.
- No interactúa directamente con los usuarios finales.

La arquitectura Cliente-Servidor empleada en FIRMA dota a la aplicación de las siguientes ventajas:

- **Escalabilidad:** se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- **Fácil mantenimiento:** al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de Cliente-Servidor que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.

Pero también se pueden encontrar una serie de inconvenientes derivados de esta arquitectura:

- La congestión del tráfico ha sido siempre un problema en el paradigma de Cliente-Servidor. Cuando una gran cantidad de clientes envían peticiones simultaneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor).
- Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico,

sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste.

3.3 Diagramas de casos de uso

Es necesario realizar la definición de los actores que van a intervenir en los diferentes casos de uso de la aplicación. En el caso de la aplicación FIRMA, y debido a su arquitectura se han establecido dos actores diferenciados.

En el lado cliente, se encuentra el **Usuario,** que es el actor que interactúa de forma directa con la aplicación y por último, en el lado del servidor, se halla el actor **Administrador Servidor,** éste exclusivamente interviene en el caso que sea necesario realizar algún tipo de cambio en la configuración del Servidor FIRMA.

En los diagramas de casos de uso mostrados a continuación se puede apreciar la funcionalidad del sistema desde un punto de vista externo para cada una de las dos partes de FIRMA:

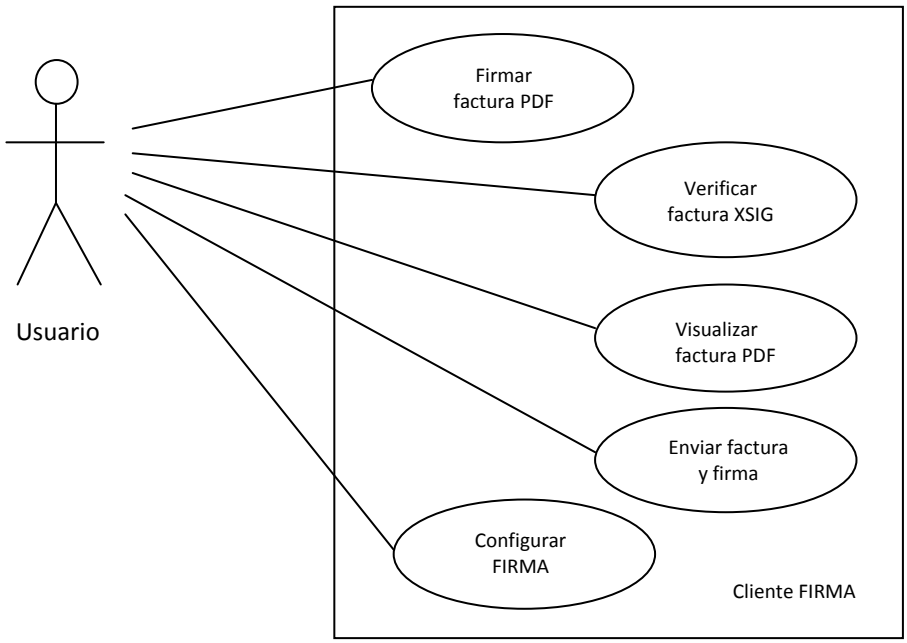

Ilustración 10: Diagrama de casos de uso Cliente FIRMA

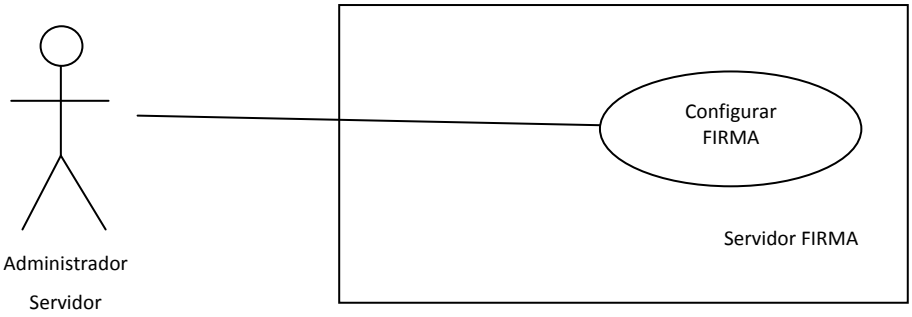

Ilustración 11: Diagrama de casos de uso Servidor FIRMA

A continuación se explican los diferentes casos de uso que existen:

Cliente FIRMA:

Nombre caso de uso:	Firmar factura PDF
Actores:	Usuario
Objetivo:	Firmar la factura contenida en el fichero PDF
Precondiciones:	-
Postcondiciones:	Factura firmada (obtención del fichero de firma XSIG)
Escenario básico:	Abrir factura PDF. Firmar factura: • Enviar factura PDF al Servidor FIRMA. • Recibir factura en formato Facturae. • Introducir la contraseña de la tarjeta inteligente. • Elegir el certificado para firmar. • "Continuar".

Tabla 5: Especificación textual caso de uso: Firmar factura PDF

Nombre caso de uso:	Verificar factura XSIG
Actores:	Usuario
Objetivo:	Verificar la autenticidad e integridad de la factura y su firma
Precondiciones:	-
Postcondiciones:	Mensaje de resultado
Escenario básico:	Abrir fichero de firma XSIG. Verificar firma: • Enviar fichero firma XSIG al Servidor FIRMA. • Recibir resultado de la comprobación. • Notificar resultado al usuario.

Tabla 6: Especificación textual caso de uso: Verificar factura

Nombre caso de uso:	Visualizar factura PDF
Actores:	Usuario
Objetivo:	Mostrar fichero factura PDF y dotar al usuario de ciertas funcionalidades (imprimir, ampliar vista).
Precondiciones:	-
Postcondiciones:	Visualización de la factura
Escenario básico:	Abrir factura PDF. Operar con el contenido del fichero: • Imprimir. • Ampliar zoom. • Alejar zoom.

Tabla 7: Especificación textual caso de uso: Visualizar factura

Nombre caso de uso:	Enviar factura y firma
Actores:	Usuario
Objetivo:	Enviar por correo electrónico la factura (PDF) y su firma (XSIG)
Precondiciones:	Factura PDF firmada
Postcondiciones:	Factura y firma enviadas por correo electrónico
Escenario básico:	Enviar por correo. Introducir los datos del correo electrónico: • Para. • Con copia (opcional). • Con copia oculta (opcional). • Asunto (opcional). • Mensaje (opcional). • “Enviar”.

Tabla 8: Especificación textual caso de uso: Enviar factura y firma

Nombre caso de uso:	Configurar FIRMA
Actores:	Usuario
Objetivo:	Configurar los parámetros de correo electrónico y los datos del Servidor FIRMA para el correcto funcionamiento de la aplicación.
Precondiciones:	Conocer los datos del correo y del Servidor FIRMA
Postcondiciones:	Cliente FIRMA configurado

Escenario básico:	“Opciones/Configurar”:
	• Introducir datos del Servidor FIRMA:
	  ○ IP del servidor.
	  ○ Puerto.
	• Introducir datos del correo electrónico:
	  ○ Nombre o Compañía para indicar en el remitente del correo.
	  ○ Correo electrónico a través del cual enviar el correo.
	  ○ Contraseña y confirmación.
	  ○ Servidor de correo a utilizar.

	○ Puerto del servidor de correo. ○ Indicar si requiere autenticación. • "Aceptar".

Tabla 9: Especificación textual caso de uso: Configurar FIRMA (Cliente)

Servidor FIRMA:

Nombre caso de uso:	Configurar FIRMA
Actores:	Administrador Servidor
Objetivo:	Configurar el número de puerto en el que se ejecuta el Servidor FIRMA para el correcto funcionamiento de la aplicación.
Precondiciones:	-
Postcondiciones:	Servidor FIRMA configurado
Escenario básico:	Editar el fichero **/config/server.properties**: • Establecer el valor deseado para el puerto en el parámetro **`port_xsig`**. • Guardar los cambios del fichero.

Tabla 10: Especificación textual caso de uso: Configurar FIRMA (Servidor)

3.4 Diagramas de actividad

A continuación se muestran los diagramas de actividad para cada uno de los anteriores casos de uso que han sido expuestos. Al igual que en los diagramas de casos de uso, se han divido en dos partes, Cliente FIRMA y Servidor FIRMA.

Cliente FIRMA:

En primer lugar, en la Ilustración 12, se muestra el diagrama de actividad para el caso de uso Firmar factura PDF.

Primeramente se seleccionará el fichero PDF con la factura que se desea firmar, a continuación la aplicación mostrará la factura al usuario y éste, si lo considera necesario, puede ejecutar la opción de **SALIR** con lo que la actividad concluye, puede realizar ciertas operaciones, como son por ejemplo, imprimir el documento, alejarlo, acercarlo, todas estas opciones aparecen recogidas en el diagrama como **OTRAS,** y por último, puede elegir **FIRMAR**.

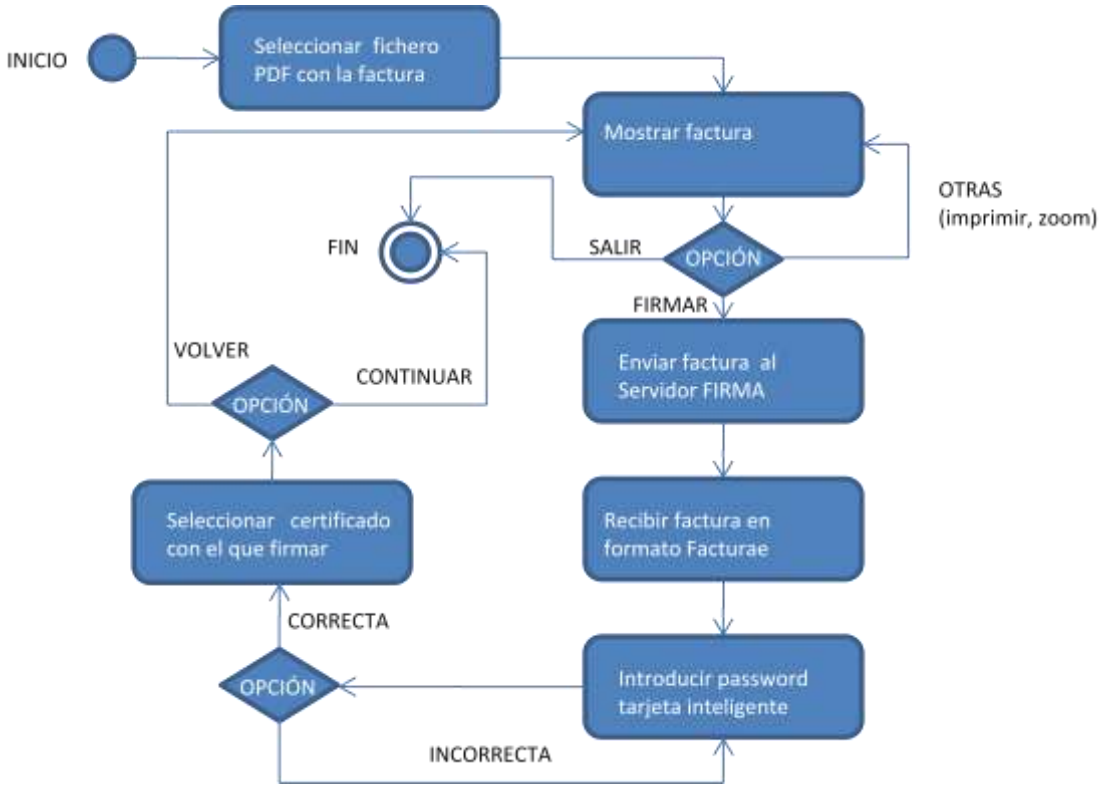

Ilustración 12: Diagrama de actividad: Firmar factura PDF

Cuando el usuario haya elegido **FIRMAR,** la aplicación envía la factura al Servidor para que éste la transforme al formato del esquema Facturae versión 3.1, y cuando esté lista recibe la factura en dicho formato en un fichero XSIG.

Posteriormente, se solicita al usuario la contraseña de la tarjeta inteligente insertada en el lector, ya sea, DNIe o la tarjeta criptográfica de la FNMT-RCM, por poner unos ejemplos, en el caso de que la contraseña sea errónea será solicitada de nuevo, en caso contrario, se le mostrará al usuario una ventana de selección de certificados para que el usuario elija el certificado que quiere realizar la firma de la factura.

Llegado este momento, el usuario tiene dos opciones: **VOLVER,** entonces la firma no se lleva a cabo y se vuelve a visualizar la factura, o bien, **CONTINUAR,** se lleva a cabo la firma de la factura, obteniendo como resultado un fichero XSIG de firma que contiene la información de la factura PDF y su firma digital con el certificado empleado.

La Ilustración 13, muestra el diagrama de actividad del caso de uso Verificar factura XSIG.

En primer lugar, una que vez el usuario quiere comprobar la autenticidad e integridad de una factura el primer paso es seleccionar el fichero de firma XSIG de esa factura.

Posteriormente, la aplicación Cliente de FIRMA envía dicho fichero al Servidor FIRMA para que proceda con su verificación. Una vez ésta ya ha concluido, el Cliente FIRMA recibe su resultado y se lo facilita al usuario.

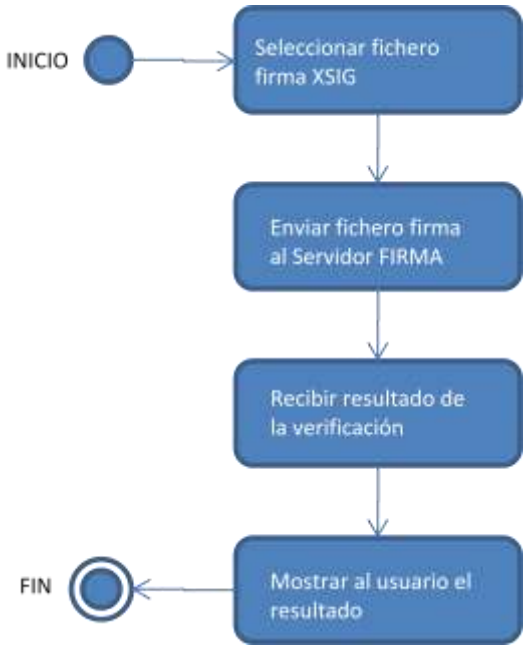

Ilustración 13: Diagrama de actividad: Verificar factura XSIG

En la Ilustración 14, se muestra el diagrama de actividad para el caso de uso Visualizar factura PDF.

En primer lugar, se seleccionará la factura PDF que se desea visualizar. Posteriormente una vez mostrada el contenido del fichero PDF, el usuario tendrá dos opciones: **OPERACIÓN**, lo que le permitirá, por ejemplo, imprimir el documento, realizar zoom sobre el mismo, y **OTRAS,** que permitirán al usuario cierta funcionalidad como firmar la factura que se está mostrando o salir del visor, pero estas funcionalidades ya son externas al diagrama de actividad del caso de uso Visualizar factura PDF.

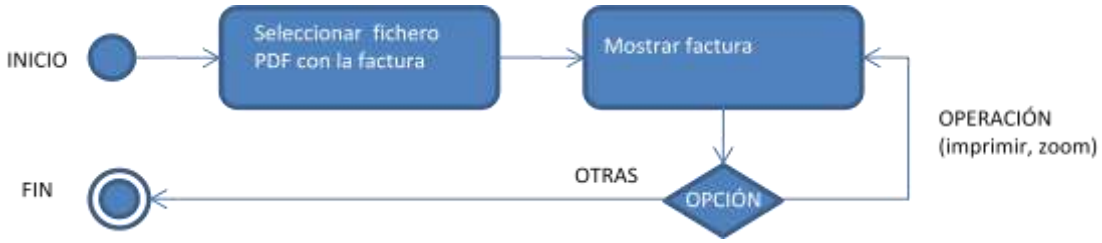

Ilustración 14: Diagrama de actividad: Visualizar factura PDF

La Ilustración 15, muestra el diagrama de actividad del caso de uso Enviar factura y firma.

En primer lugar, hay que firmar la factura PDF. Como puede apreciarse en el diagrama, **Firmar factura PDF,** se considera en este caso de uso como una subactividad, que está completamente desarrollada en el diagrama de actividad para el caso de uso Firmar factura PDF, ver Ilustración 12.

Una vez firmada la factura, el usuario tiene dos opciones: en primer lugar, **SALIR,** y como consecuencia, no se produce el envío por correo electrónico de la factura y su firma, y en segundo lugar **ENVIAR POR MAIL**.

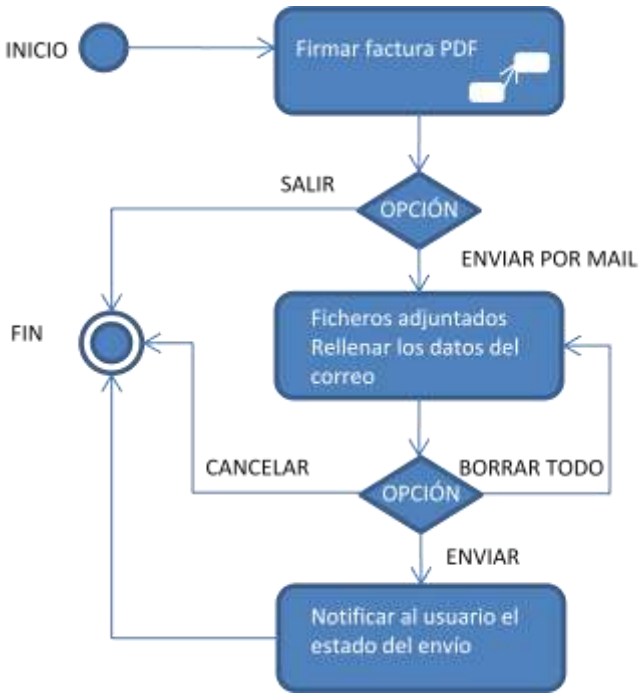

Ilustración 15: Diagrama de actividad: Enviar factura y firma

Una vez que el usuario ha elegido **ENVIAR POR MAIL,** la aplicación adjunta de forma automática tanto el fichero PDF que contiene la factura como el fichero de firma XSIG al correo electrónico. Los datos necesarios restantes para poder realizar el envío por correo electrónico, como son, por ejemplo, el destinatario, las copias, el asunto y el mensaje son solicitados al usuario mediante un formulario. En el que el usuario tiene tres opciones que puede llevar a cabo: **CANCELAR,** se cancela el envío, **BORRAR TODO,** permite al usuario tener en blanco todo el formulario de nuevo y, por último, **ENVIAR,** se produce el envío.

Para finalizar, se notifica al usuario el estado del envío que acaba de realizar.

En la Ilustración 16, se muestra el diagrama de actividad para el caso de uso Configurar FIRMA.

En primer lugar el usuario debe seleccionar la pestaña correspondiente en el menú **Opciones/Configurar** según los datos que desee configurar: **DATOS SERVIDOR** o bien, **DATOS CORREO.**

Si seleccionó **DATOS SERVIDOR,** el usuario introducirá los datos necesarios referentes al Servidor de FIRMA tales como la IP y el puerto en el que recibe las peticiones dicho Servidor. Posteriormente en el caso de que alguno de estos datos sean erróneos serán solicitados de nuevo al usuario, en caso contrario, el usuario elegirá si desea configurar los datos referentes al correo electrónico, **DATOS CORREO,** o bien terminar con la configuración mediante **ACEPTAR.**

Si seleccionó **DATOS CORREO,** el usuario facilitará a la aplicación mediante un formulario los datos necesarios para la correcta configuración del correo electrónico, como el nombre o compañía que se utilizará como remitente del correo, la dirección de correo con la que enviar los mails, la contraseña y su confirmación, el servidor y el puerto de correo utilizado para el envío e indicar si la cuenta de correo requiere autenticación. Posteriormente en el caso de que alguno de estos datos sean erróneos serán solicitados de nuevo al usuario, en caso contrario, el usuario elegirá si desea configurar los datos referentes al Servidor FIRMA, **DATOS SERVIDOR,** o bien terminar con la configuración mediante **ACEPTAR.**

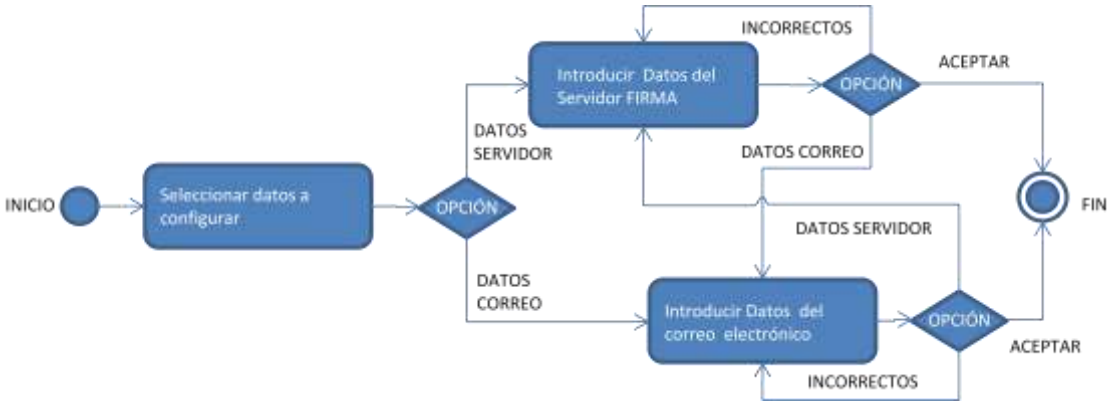

Ilustración 16: Diagrama de actividad: Configurar FIRMA (Cliente)

Servidor FIRMA:

La Ilustración 17, muestra el diagrama de actividad del caso de uso Configurar FIRMA.

En primer lugar, es necesario editar con un editor de texto, por ejemplo, el bloc de notas el fichero **server.properties** contenido en el directorio **/config.**

Seguidamente se cambia el valor del parámetro **`port_xsig`** por el nuevo valor del puerto en el que el Servidor FIRMA escuchará las peticiones de los clientes.

Finalmente, se guardan los cambios realizados en el fichero **server.properties.**

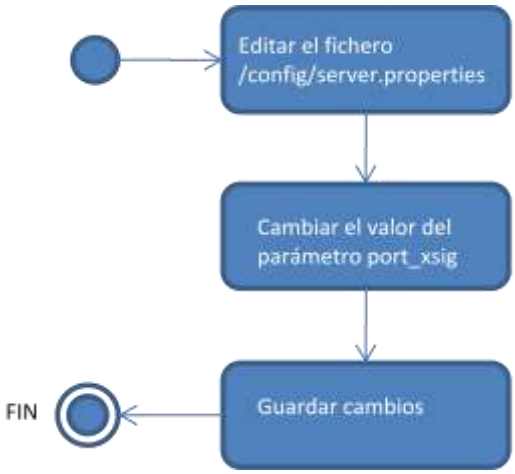

Ilustración 17: Diagrama de actividad: Configurar FIRMA (Servidor)

3.5 Diseño factura PDF

FACTURA COMPLETA

Número de factura / Invoice number
2009-39
Fecha de expedición de la factura (dd-mm-aaaa) / Invoice date
25-02-2009

Emisor (nombre, dirección...) / Seller (name, address, tax reference)	CIF / NIF del Emisor / Sellers reference	Contacto / Contact
Persona Jurídica / Persona Física Company Computer SA C/ Mayour 33 1 5º E 28001 Argamasilla de Alba Ciudad Real ESP	A82735122	Rubén Pérez Pérez
	Teléfono / Telephone	Fax / Fax
	926989900	926989901
	Correo electrónico / Email	Dirección web / Web
	contacta@comp.es	www.company-computer.es

Receptor (nombre, dirección...) / Buyer (name, address...)	CIF / NIF del Receptor / Buyer reference	Contacto / Contact
Persona Jurídica / Persona Física Bakgammon Games SA C/ Mayor 211 28001 Madrid Madrid ESP	A99887723	Ángel López López
	Teléfono / Telephone	Fax / Fax
	916998800	916998801
	Correo electrónico / Email	Dirección web / Web
	contacto@games.es	www.backgammon.es

Fecha de entrega (dd-mm-aaaa) / Delivery date	Divisa de pago / Transaction currency
26-02-2009	EUR

Código Detalle/ Item/ Packages	Descripción detalle / Item description	Cantidad/ Quantity	Precio unitario /Unit price	Porcentaje del descuento/ Discount percentage	Total sin impuestos/ Amount (gross)	Tipo de impuesto / Tax	Porcentaje del Impuesto/Tax percentage	Total con impuestos/ Amount Due
1	Miniordenador	2,0	230,0000	5,0000	437,00	IVA	16,00	506,92
2	Portátil	2,0	120,0000	5,0000	228,00	IRPF	16,00	264,48
3	Impresora	2,0	100,0000	5,0000	190,00	ICIO	16,00	220,40

Importe total sin Impuestos/ Total Ammount Due	
	855,00
Descuentos / Discounts	
D. Globales-5,00%	42,75
Cargos / Charges	
Gtos de envio+10%	85,50
Importe total con Impuestos/ Total Ammount Due	
	1034,55

Ilustración 18: Muestra de factura en fichero PDF

En la ilustración anterior se puede visualizar un ejemplo del formato de la factura empleado para este Proyecto. Este formato será el que soporte la aplicación FIRMA para las facturas contenidas en ficheros PDF.

Como puede apreciarse están recogidos todos los campos que obligatoriamente debe contener toda factura mencionados en el apartado *2.4.1 Definición (Factura electrónica).* Los campos Contacto, Fax, Web y Email tanto del Emisor como Receptor son opcionales.

3.6 Diseño firma XSIG

A continuación se expone el formato que presenta un fichero de firma XSIG, cuyo contenido es la información de la factura PDF correspondiente y su firma digital. El fichero de firma XSIG sigue el esquema de Facturae versión 3.1.

Para que pueda resultar más fácil su comprensión, se va a ir detallando cada uno de los bloques que conforman el esquema Facturae versión 3.1, ya explicados cada uno de ellos en el apartado *2.4.2 Formato Facturae, estándar para las facturas electrónicas a nivel nacional*, mediante una factura de ejemplo utilizada para la realización de este Proyecto e indicando la información que contiene en cada uno de ellos:

Primer bloque, *FileHeader*

Corresponde a la cabecera del fichero XML que representa a la factura.

En él, podemos encontrar la siguiente información:

- Versión de Facturae empleada. (1)
- Modalidad de la factura, *Individual* o *Lote*. (2)
- Actor que firma la factura, *Emisor, Receptor* o *Tercero*, si es éste último, indica que hay un caso de subfacturación, y por lo tanto, también se encontraría sus datos identificativos, como en el ejemplo, no nos encontramos en este caso, no aparece. (3)

- Lote. Información referente al lote de facturas contenidas en el fichero, en nuestro ejemplo, el fichero sólo dispone de una factura. (4)
- Datos cesión factoring. Es opcional, en nuestro ejemplo, no procede.

```
<?xml version="1.0" encoding="UTF-8"?>
<fe:Facturae xmlns:fe="http://www.facturae.es/Facturae/2007/v3.1/Facturae"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

<FileHeader>
     <SchemaVersion>3.1</SchemaVersion>                                    } (1)

     <Modality>I</Modality>                                                } (2)

     <InvoiceIssuerType>EM</InvoiceIssuerType>                             } (3)

     <Batch>
          <BatchIdentifier>A82735122392010-</BatchIdentifier>
          <InvoicesCount>1</InvoicesCount>
          <TotalInvoicesAmount>
               <TotalAmount>3100.20</TotalAmount>
          </TotalInvoicesAmount>
          <TotalOutstandingAmount>                                         } (4)
               <TotalAmount>3100.20</TotalAmount>
          </TotalOutstandingAmount>
          <TotalExecutableAmount>
               <TotalAmount>3100.20</TotalAmount>
          </TotalExecutableAmount>
          <InvoiceCurrencyCode>EUR</InvoiceCurrencyCode>
     </Batch>
</FileHeader>
```

Tabla 11: Fragmento factura de ejemplo Facturae (Bloque *FileHeader*)

Segundo bloque, *Parties*

Este bloque reúne toda la información referente tanto al Emisor como al Receptor de la factura, tal y como se puede apreciar en el siguiente fragmento de la factura de ejemplo se halla la siguiente información (de forma duplicada, para el Emisor y el Receptor):

- NIF, o en su defecto, CIF. (1)
- Nombre o Compañía, junto con su correspondiente domicilio. (2)
- Teléfono y fax, este último, es un dato opcional, aunque nuestro ejemplo dispone de él. (3)
- Email y web, también son opcionales y en este caso disponemos de ellos en el ejemplo. (4)
- Persona de contacto, opcional, aunque nuestro ejemplo si lo tiene. (5)

```
<Parties>
   <SellerParty>
      <TaxIdentification>
          <PersonTypeCode>J</PersonTypeCode>
          <ResidenceTypeCode>R</ResidenceTypeCode>
          <TaxIdentificationNumber>A82735122</TaxIdentificationNumber>          (1)
      </TaxIdentification>
      <LegalEntity>
          <CorporateName>Company Computer SA</CorporateName>
          <AddressInSpain>
             <Address>C/ Mayour 33 1 5° E</Address>
             <PostCode>28001</PostCode>                                         (2)
             <Town>Argamasilla de Alba</Town>
             <Province>Ciudad Real</Province>
             <CountryCode>ESP</CountryCode>
          </AddressInSpain>
          <ContactDetails>
             <Telephone>926989900</Telephone>                                   (3)
             <TeleFax>926989901</TeleFax>
             <WebAddress>www.company-computer.es</WebAddress>                   (4)
             <ElectronicMail>contacta@comp.es</ElectronicMail>
             <ContactPersons>Rubén Pérez Pérez</ContactPersons>                 (5)
          </ContactDetails>
      </LegalEntity>
   </SellerParty>

   <BuyerParty>
      <TaxIdentification>
         <PersonTypeCode>J</PersonTypeCode>
         <ResidenceTypeCode>R</ResidenceTypeCode>
         <TaxIdentificationNumber>A99887700</TaxIdentificationNumber>           (1)
      </TaxIdentification>
      <LegalEntity>
         <CorporateName>Burbuja.com SA</CorporateName>
         <AddressInSpain>
            <Address>C/ Derribo 215</Address>
            <PostCode>08001</PostCode>                                          (2)
            <Town>Barcelona</Town>
            <Province>Barcelona</Province>
            <CountryCode>ESP</CountryCode>
         </AddressInSpain>
         <ContactDetails>
            <Telephone>936998800</Telephone>                                    (3)
            <TeleFax>936998801</TeleFax>
            <WebAddress>www.burbuja.com</WebAddress>                            (4)
            <ElectronicMail>contacto@burbuja.com</ElectronicMail>
            <ContactPersons>Roger Montezuma</ContactPersons>                    (5)
         </ContactDetails>
      </LegalEntity>
   </BuyerParty>
</Parties>
```

Tabla 12: Fragmento factura de ejemplo Facturae (Bloque *Parties*)

Tercer bloque, *Invoices*

Representa el conjunto de facturas que contiene el fichero. En nuestro ejemplo, el fichero está formado por una sola factura, por lo tanto, este bloque está formado por dicha

factura, su información en conjunto y su información detallada para cada elemento facturado en la que hay que destacar:

- Número de factura, fecha de expedición. (1)
- Impuestos totales sobre la base imponible total de la factura. (2)
- Totales de la factura, descuentos y cargos. (3)
- Información sobre cada unos de los elementos facturados, en nuestro caso, sólo hay uno. (4)
- Información adicional de la factura, se han utilizado estos campos disponibles según el esquema Facturae, para almacenar en ellos tanto el nombre del fichero de la factura en formato PDF como su hash, requisitos imprescindibles para la correcta implementación de este Proyecto, para poder establecer una relación inequívoca entre la factura PDF y su firma (fichero XSIG). (5)

```
<Invoices>
   <Invoice>
      <InvoiceHeader>
         <InvoiceNumber>39</InvoiceNumber>
         <InvoiceSeriesCode>2010-</InvoiceSeriesCode>
         <InvoiceDocumentType>FC</InvoiceDocumentType>
         <InvoiceClass>OO</InvoiceClass>
      </InvoiceHeader>
      <InvoiceIssueData>                                          (1)
         <IssueDate>2010-03-14</IssueDate>
         <InvoiceCurrencyCode>EUR</InvoiceCurrencyCode>
         <TaxCurrencyCode>EUR</TaxCurrencyCode>
         <LanguageName>es</LanguageName>
      </InvoiceIssueData>
      <TaxesOutputs>
         <Tax>
            <TaxTypeCode>01</TaxTypeCode>
            <TaxRate>16.00</TaxRate>
            <TaxableBase>                                         (2)
               <TotalAmount>2562.15</TotalAmount>
            </TaxableBase>
            <TaxAmount>
               <TotalAmount>409.94</TotalAmount>
            </TaxAmount>
         </Tax>
      </TaxesOutputs>
      <InvoiceTotals>
         <TotalGrossAmount>2562.15</TotalGrossAmount>
         <GeneralDiscounts>
            <Discount>                                            (3)
               <DiscountReason>D. Globales</DiscountReason>
               <DiscountRate>5.0000</DiscountRate>
               <DiscountAmount>128.11</DiscountAmount>
            </Discount>
         </GeneralDiscounts>
         <GeneralSurcharges>
            <Charge>
               <ChargeReason>Gtos de envío</ChargeReason>
```

```
               <ChargeRate>10.0000</ChargeRate>
               <ChargeAmount>256.22</ChargeAmount>
            </Charge>
         </GeneralSurcharges>
         <TotalGeneralDiscounts>128.11</TotalGeneralDiscounts>
         <TotalGeneralSurcharges>256.22</TotalGeneralSurcharges>
         <TotalGrossAmountBeforeTaxes>
            2690.26
         </TotalGrossAmountBeforeTaxes>
         <TotalTaxOutputs>409.94</TotalTaxOutputs>
         <TotalTaxesWithheld>0.00</TotalTaxesWithheld>
         <InvoiceTotal>3100.20</InvoiceTotal>
         <TotalOutstandingAmount>3100.20</TotalOutstandingAmount>
         <TotalExecutableAmount>3100.20</TotalExecutableAmount>
      </InvoiceTotals>                                                    (3)
      <Items>
         <InvoiceLine>
            <ItemDescription>1 Server_X2000</ItemDescription>
            <Quantity>3.00</Quantity>
            <UnitPriceWithoutTax>899.000000</UnitPriceWithoutTax>
            <TotalCost>2697.00</TotalCost>
            <DiscountsAndRebates>
               <Discount>
                  <DiscountReason>Motivo por el que se aplica descuento
                  </DiscountReason>
                  <DiscountRate>5.0000</DiscountRate>
                  <DiscountAmount>134.85</DiscountAmount>
               </Discount>
            </DiscountsAndRebates>
            <GrossAmount>2562.15</GrossAmount>
            <TaxesOutputs>
               <Tax>
                  <TaxTypeCode>01</TaxTypeCode>
                  <TaxRate>16.00</TaxRate>
                  <TaxableBase>
                     <TotalAmount>2562.15</TotalAmount>
                  </TaxableBase>
                  <TaxAmount>
                     <TotalAmount>409.94</TotalAmount>
                  </TaxAmount>
               </Tax>
            </TaxesOutputs>
         </InvoiceLine>                                                   (4)
      </Items>
      <AdditionalData>
         <RelatedInvoice>2010_03_14_Computer_2010_39.pdf</RelatedInvoice>
         <RelatedDocuments>
            <Attachment>
               <AttachmentFormat>pdf</AttachmentFormat>
               <AttachmentData>
1cfee104e9bf5e1f38be4d0fb7a160ae321165fcdb
6ee22b16eca5f9821e2a5efff716b07c792a58158c
300a987e82d9da12d3734421563b8f21731c2c326c51
               </AttachmentData>
            </Attachment>
         </RelatedDocuments>
      </AdditionalData>                                                   (5)
   </Invoice>
</Invoices>
```

Tabla 13: Fragmento factura de ejemplo Facturae (Bloque *Invoices*)

Cuarto bloque, *Extensions*

Como se ha mencionado en el apartado *2.4.2 Formato Facturae, estándar para las facturas electrónicas a nivel nacional*, este bloque es opcional y en él se podrían incorporarse nuevas definiciones estructuradas cuando sean de interés conjunto para emisores y receptores, y no estén ya definidas en el esquema de la factura. En nuestro ejemplo no procede.

Quinto bloque, *dsSignature*

Este bloque contiene el conjunto de datos asociados a la factura que garantizarán la autoría y la integridad del mensaje. Está definido como opcional para facilitar la verificación y el tránsito del fichero. No obstante, debe cumplimentarse este bloque de firma electrónica para que se considere una factura electrónica válida legalmente frente a terceros.

```
<ds:Signature xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" Id="Signature">
   <ds:SignedInfo Id="Signature-SignedInfo">
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference Id="SignedPropertiesID"
Type="http://uri.etsi.org/01903#SignedProperties" URI="#Signature-SignedProperties">
         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
         <ds:DigestValue>3eQJPfjUEH9UFrZvGJKqEPc/0zE=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="">
         <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
         </ds:Transforms>
         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
         <ds:DigestValue>LuZgCNnUJnr4N9lYo227NCkwmOQ=</ds:DigestValue>
       </ds:Reference>
      <ds:Reference URI="#Certificate1">
         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
         <ds:DigestValue>IXoqQlmoJdndO2FEiN+fXShQub4=</ds:DigestValue>
      </ds:Reference>
   </ds:SignedInfo>
   <ds:SignatureValue Id="SignatureValue">
hMR1AO582gsBUt7PMuoe7TCvv/bCMGvbseCGsXH6b+VJSHCubYfvG30ZDPcFYFAL82/OSbv1Zxnr
cl8P/KmNY5iOjKE+dEMOWjB8I911K91xbHo9F+it9bNNWrWsO1fBalh/3mm5yaVsE3Sbz6b+3VSM
gMnqcz4+1LqhxIWV3/bpNvbODeHfPa2rTOXi4O7uSdCXlYt14ecPyecwyzQ6hm9Aa5P7BVGhEsqT
nWfmKzXn3r9D7msVYFnzMoUfGOJEAoJDHtasBy08d78lLhaIhABfX92cVoWWC5AFNpgmFWF8bsA8
5kgNGhIUpsrAKYkHj4dBlIy+3uoojMgKNeeRGw==
   </ds:SignatureValue>
   <ds:KeyInfo Id="Certificate1">
      <ds:X509Data>
         <ds:X509Certificate>
MIIFzzCCBLegAwIBAgIERFQpWzANBgkqhkiG9w0BAQUFADBcMQswCQYDVQQGEwJFUzEoMCYGA1UE
CgwfRE1SRUNDSU9OIEdFTkVSQUwgREUgTEEgUE9MSUNJQTENMAsGA1UECwwERE5JRTEUMBIGA1UE
AwwLQUMgRE5JRSAwMDMwHhcNMDkwMjE2MTEwMTQ5WhcNMTEwODE2MTAzMTQ3WjBtMQswCQYDVQQG
EwJFUzESMBAGA1UEBRMJNDkwMjAyMjdNMQ8wDQYDVQQEDAZHQUxWRVoxEDAOBgNVBCoMB0VTVEVC
QU4xJzAlBgNVBAMMHkdBTFZFWiBNQVJUSU4sIEVTVEVCQU4gKEZJUk1BKTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBALQU6Rl5EU5zi200HGCHPBTSAZa3MtgCuVP68DWDI08zHKqZ4ZAZ
YzSFUmnwQ0xmpjN7dyqFtEumn31jEbGeq0yBp3evNYy+JYI/3hoUfFflxSh+3CTXGLG3yYFJ5y7n
```

```
n/LnLRBrgiHC8WNMiyYydos7jkv2pizSP8gZFyOGIdujA52mgQ8otuq7+auXJENiwR2aZwTdjzPJ
/XK67RrOBYXU+yVVYlfI2Qvjku60E/prt5+biHL9BKm0k33QUpyGdEycOmihYYMBGS+kwyhgthto
8kdRl3QFYnjcdWiYPxymrXoJMkdZuxM+euDyVsmDDzwAlMS9rL0LNM2uTmdMLdMCAwEAAaOCAoYw
ggKCMA4GA1UdDwEB/wQEAwIGQDAoBgNVHQkEITAfMB0GCCsGAQUFBwkBMREYDzE5ODUwMTMwMTIw
MDAwWjBCBghghVQBAgIEAQQ2MDQwMgIBAjALBglghkgBZQMEAgEEII99ytB7WdVle04h/ky8yimq
XqOWL9dJ7MqGZrhPwuM5MIHwBggrBgEFBQcBAgSB4zCB4DAyAgEBMAsGCWCGSAFlAwQCAQQgNnlX
dZaS++rsLOCI8feHqDhFk3oubNAlBGlkv0qyYqwwMgIBADALBglghkgBZQMEAgEEIFD/0uJWgwze
UQUOYxYhkmxKxHM6OyAPrq7AAj5ZWdVhMDoGCWCFVAECAgQCATALBglghkgBZQMEAgEEINSKhzhy
NrpqWd+I3VOgAr++Lz9jKBpH/vMmpql27m8ZMDoGCWCFVAECAgQCBjALBglghkgBZQMEAgEEIL7n
Vazj5cS87gbY9yKO6YrGOzzLncZjuZjDSNXwA2cUMAwGA1UdEwEB/wQCMAAwIgYIKwYBBQUHAQME
FjAUMAgGBgQAjkYBATAIBgYEAI5GAQQwYAYIKwYBBQUHAQEEVDBSMB8GCCsGAQUFBzABhhNodHRw
Oi8vb2NzcC5kbmllLmVzMC8GCCsGAQUFBzAChiNodHRwOi8vd3d3LmRuaWUuZXMvY2VydHMvQUNS
YWl6LmNydDA7BgNVHSAENDAyMDAGCGCFVAECAgIDMCQwIgYIKwYBBQUHAgEWFmh0dHA6Ly93d3cu
ZG5pZS5lcy9kcGMwHwYDVR0jBBgwFoAUA5VDQHSI1GFWhU6bunxAVPBPou4wHQYDVR0OBBYEFECZ
nOiBSe8+1nmDhCSSPiBNLTyCMA0GCSqGSIb3DQEBBQUAA4IBAQAFKoBVMaf0sI7ABdchRx5YDXYq
MvttMr2vZ3l2qEzNUvN2MBbAqPz+/HKrT1ceOoavHWXecfc3aIO48C/HJb7kAjXTeclmPqLH1RHa
4USfyASHIEPfDaldOX+GCofVmsA0peTqy/caHyIs5c7ccjI7qTUPG0BW9WGS4l2RVlR5xuEhDTZ7
+44cYuqquxwp6lmRbJAc7uWjvXrAb7LuNordB909aoVKZW8amNxmW9SorG6Isfu6iTHbfcB7RnCS
Rd234CgcEMLT/a8idS8EORFcdXLXU+ibVS7i+ETXTBS/mKIfF8TFU7KYMqq9uhhWmjSuFRw7Blgl
PKyZlELI1muR</ds:X509Certificate>
      </ds:X509Data>
      <ds:KeyValue>
         <ds:RSAKeyValue>
            <ds:Modulus>
tBTpGXkRTnOLbTQcYIc8FNIBlrcy2AK5U/rwNYMjTzMcqpnhkBljNIVSafBDTGamM3t3KoW0S6af
fWMRsZ6rTIGnd681jL4lgj/eGhR8V+XFKH7cJNcYsbfJgUnnLuef8uctEGuCIcLxY0yLJjJ2izuO
S/amLNI/yBkXI4Yh26MDnaaBDyi26rv5q5ckQ2LBHZpnBN2PM8n9crrtGs4FhdT7JVViV8jZC+OS
7rQT+mu3n5uIcv0EqbSTfdBSnIZ0TJw6aKFhgwEZL6TDKGC2G2jyR1GXdAVieNx1aJg/HKategky
R1m7Ez564PJWyYMPPACUxL2svQs0za5OZ0wt0w==
            </ds:Modulus>
            <ds:Exponent>AQAB</ds:Exponent>
         </ds:RSAKeyValue>
      </ds:KeyValue>
   </ds:KeyInfo>
   <ds:Object Id="Signature-Object">
      <xades:QualifyingProperties Target="#Signature">
            <xades:SignedProperties Id="Signature-SignedProperties">
               <xades:SignedSignatureProperties>
               <xades:SigningTime>2010-03-16T19:32:31+01:00</xades:SigningTime>
               <xades:SigningCertificate>
                  <xades:Cert>
                     <xades:CertDigest>
                        <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>dlu40b3P+MIiTKq7uRreVNwieiU=</ds:DigestValue>
                     </xades:CertDigest>
                     <xades:IssuerSerial>
                        <ds:X509IssuerName>CN=AC DNIE 003,OU=DNIE,O=DIRECCION GENERAL DE
LA POLICIA,C=ES</ds:X509IssuerName>
                        <ds:X509SerialNumber>1146366299</ds:X509SerialNumber>
                     </xades:IssuerSerial>
                  </xades:Cert>
               </xades:SigningCertificate>
               <xades:SignaturePolicyIdentifier>
                  <xades:SignaturePolicyId>
                     <xades:SigPolicyId>
<xades:Identifier>http://www.facturae.es/politica de firma formato facturae/politica de
firma_formato_facturae_v3_1.pdf</xades:Identifier>
                        <xades:Description>facturae31</xades:Description>
                     </xades:SigPolicyId>
                     <xades:SigPolicyHash>
                        <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>Ohix16upD6av8N7pEvDABhEL6hM=</ds:DigestValue>
                     </xades:SigPolicyHash>
                  </xades:SignaturePolicyId>
               </xades:SignaturePolicyIdentifier>
               <xades:SignerRole>
                  <xades:ClaimedRoles>
                     <xades:ClaimedRole>emisor</xades:ClaimedRole>
                  </xades:ClaimedRoles>
               </xades:SignerRole>
```

```
            </xades:SignedSignatureProperties>
          </xades:SignedProperties>
        </xades:QualifyingProperties>
    </ds:Object>
</ds:Signature>
```

Tabla 14: Fragmento factura de ejemplo Facturae (Bloque *dsSignature*)

3.7 Diseño gráfico del sistema

A continuación se muestras varias ilustraciones que exponen el diseño gráfico de la aplicación ya implementada. Es necesario indicar que todo lo contenido en este apartado es referido a la parte Cliente de FIRMA, ya que la parte Servidor no contiene ninguna interfaz gráfica.

En primer lugar, en la Ilustración 19, se puede apreciar el diseño de la ventana principal de la aplicación:

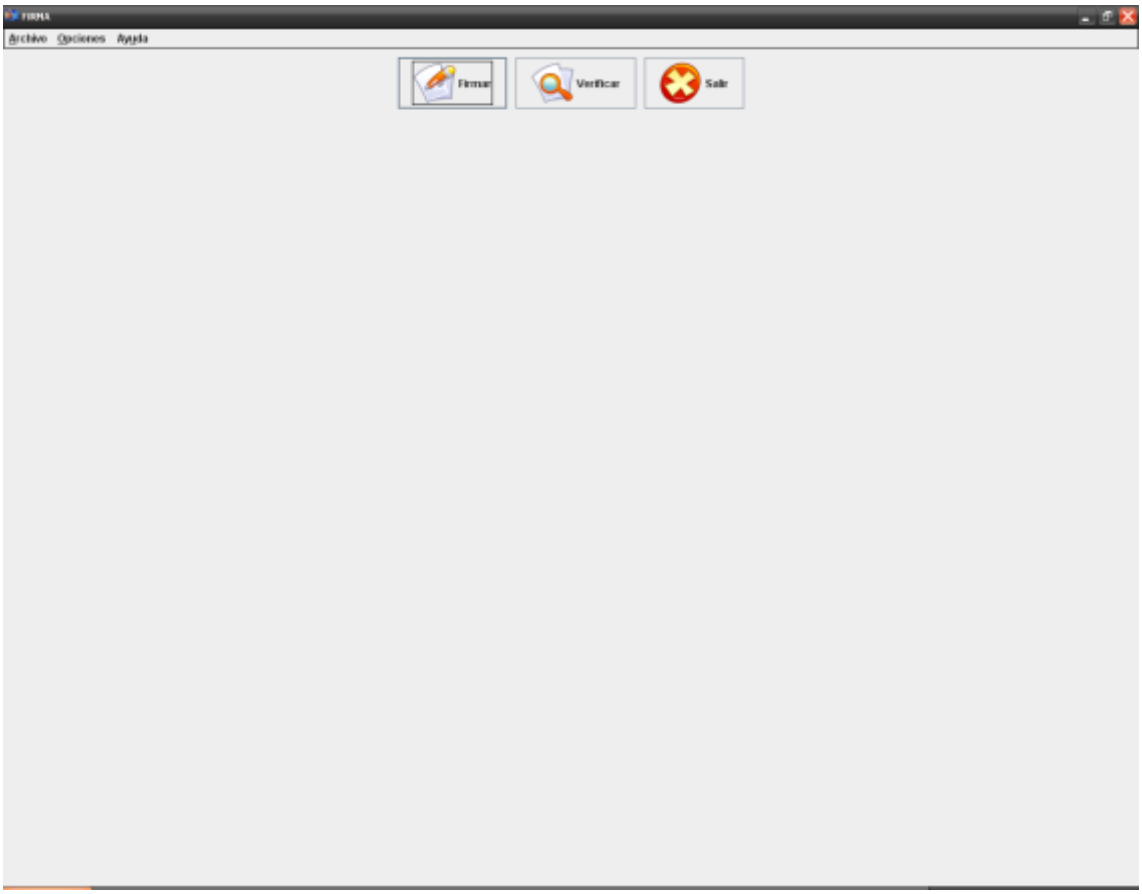

Ilustración 19: Ventana principal de la aplicación FIRMA

En la anterior ilustración se puede apreciar la existencia de una barra de menú en la que podemos encontrar los siguientes elementos:

- **Menú Archivo:**
  - **Abrir…:** Abre un cuadro de diálogo para la selección del fichero PDF que contiene la factura a tratar.
  - **Verificar firma:** Abre un cuadro de diálogo para la selección del fichero XSIG del que se desee verificar su integridad y autenticidad.

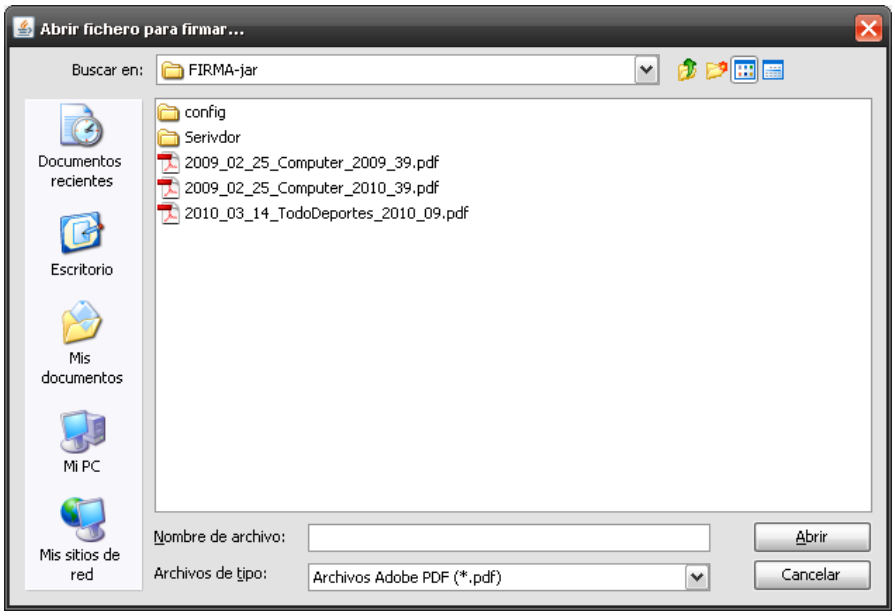

Ilustración 20: Muestra cuadro diálogo

  - **Salir:** Permite cerrar la aplicación.
- **Menú Opciones:**
  - **Configurar:** Abre la ventana de configuración donde establecer los valores necesarios a diferentes parámetros para el correcto funcionamiento de la aplicación.

Ilustración 21: Ventanas de configuración FIRMA

- **Menú Ayuda:**
  - **Acerca de FIRMA:** Para conocer más sobre FIRMA.

Ilustración 22: Acerca de FIRMA

También en la ventana principal se encuentra el panel principal en el que aparecen tres botones para facilitar al usuario su labor aunque la funcionalidad es la misma que sus elementos correspondientes del menú principal. Dichos botones son:

- **Firmar:** Igual que el elemento **Abrir…** del menú **Archivo,** ya comentado con anterioridad.
- **Verificar:** Exactamente igual que el elemento **Verificar** del menú **Archivo.**
- **Salir:** Sale de la aplicación.

A continuación, se muestra el diseño para la ventana utilizada como visor de la factura PDF:

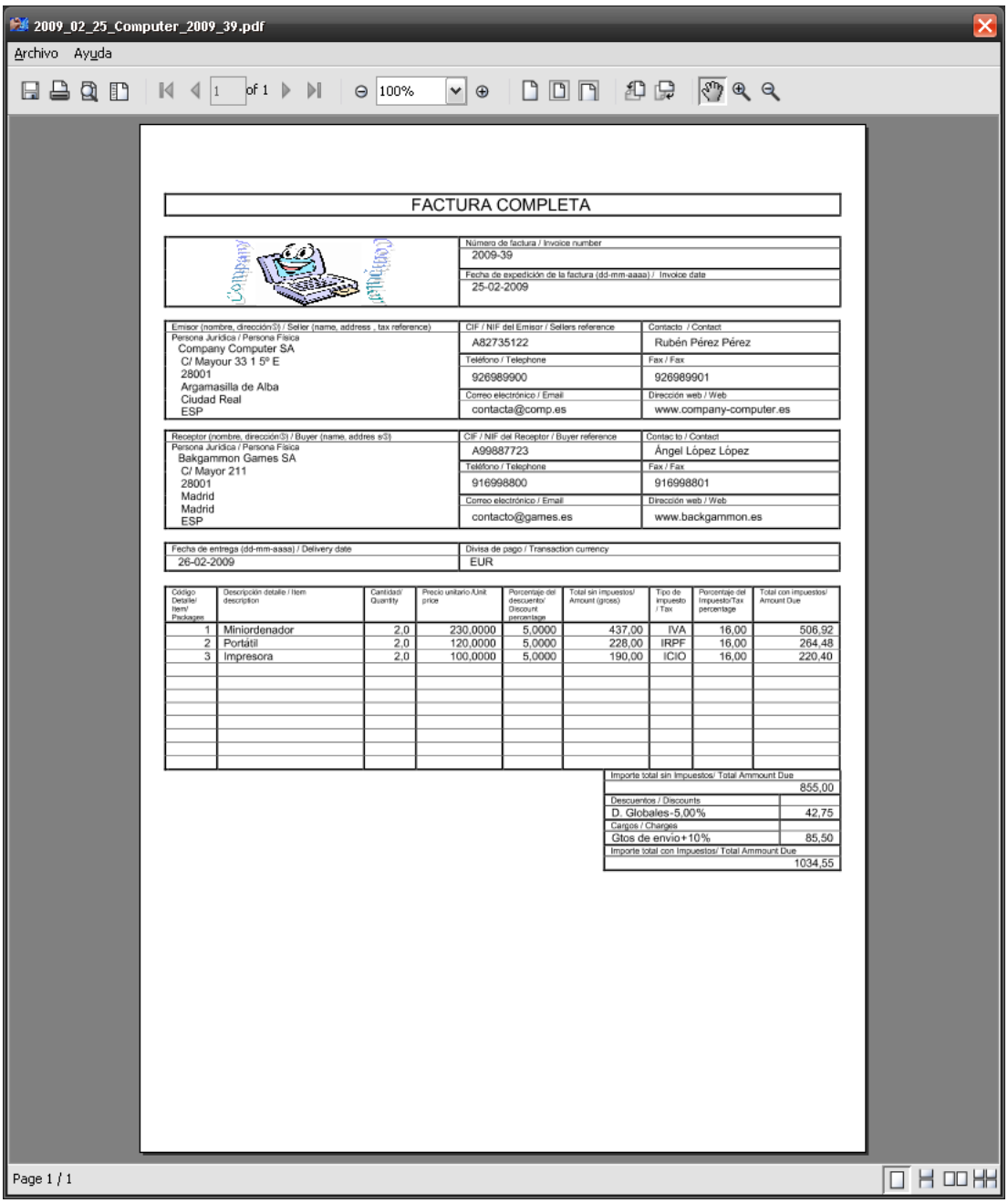

Ilustración 23: Visor de factura PDF

En la anterior ilustración se puede apreciar la existencia de una barra de herramientas que permite al usuario ciertas funciones sobre el tratamiento del fichero PDF (imprimir, guardar fichero, realizar zoom).

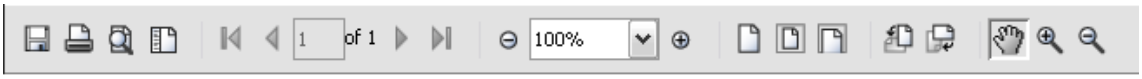

Ilustración 24: Barra de herramientas

También se puede apreciar la existencia de una barra de menú en la que podemos encontrar los siguientes elementos:

Ilustración 25: Barra de menú

- **Menú Archivo:**
  - **Firmar documento:** Se iniciará el proceso de firma, se solicitará para ello la contraseña de la tarjeta inteligente insertada en el lector y posteriormente se muestra una ventana en la que el usuario elige con que certificado contenido en la tarjeta desea firmar la factura.

Ilustración 26: Ventana contraseña DNIe

Ilustración 27: Ventana de selección de certificado

- **Enviar por mail:** Permite enviar por mail la factura PDF y su firma XSIG, una vez se ha realizado la firma de la factura.

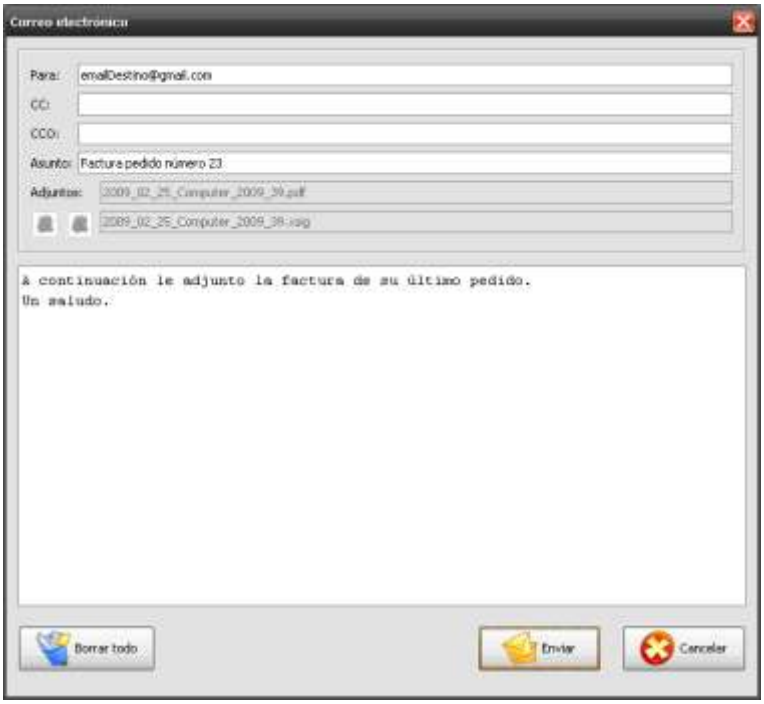

Ilustración 28: Ventana correo electrónico

- **Salir:** Se cierra la ventana.

- **Menú Ayuda:**
  - **Acerca de FIRMA:** Para conocer más sobre FIRMA.

IV. Implementación

En este aparatado se recoge toda la información relevante acerca de la implementación de este Proyecto.

Esta información está divida en dos grandes bloques principales, que se han implementado de forma independiente: Cliente FIRMA y Servidor FIRMA. Pero es necesario antes de abordar estos dos bloques conocer el procedimiento de conexión entre estas dos partes que componen la aplicación FIRMA, los sockets. Éstos permitirán la comunicación entre las dos partes que consta la aplicación.

4.1 Comunicación Cliente-Servidor

4.1.1 Sockets

Los sockets permiten implementar una arquitectura Cliente-Servidor. La comunicación ha de ser iniciada por uno de los programas que se denomina programa cliente, es decir, Cliente FIRMA. El segundo programa espera a que otro inicie la comunicación para responder peticiones, por este motivo se denomina programa servidor, Servidor FIRMA.

Un socket es un fichero existente en la máquina cliente y en la máquina servidora, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

Las propiedades de un socket dependen de las características del protocolo en el que se implementan. El protocolo utilizado en la comunicación entre el Cliente FIRMA y el Servidor FIRMA es TCP *(Transmission Control Protocol)*, aunque también es posible utilizar UDP o IPX.

Gracias al protocolo TCP, los sockets presentan las siguientes propiedades:

- Orientado a conexión.
- Se garantiza la transmisión de todos los octetos sin errores ni omisiones.
- Se garantiza que todo octeto llegará a su destino en el mismo orden en que se ha transmitido, propiedad de vital importancia para el intercambio de ficheros entre el Cliente y el Servidor FIRMA.

Estas propiedades son muy importantes para garantizar el correcto funcionamiento de los programas que tratan la información y el intercambio de ficheros, como es el caso, entre el Cliente FIRMA y el Servidor FIRMA, para que la información contenida en ellos no se vea alterada y se encuentre completa.

Posteriormente, en los dos siguientes puntos de este apartado, se puede apreciar el uso de los sockets en cada uno de los bloques que conforman la aplicación: Cliente FIRMA y Servidor FIRMA.

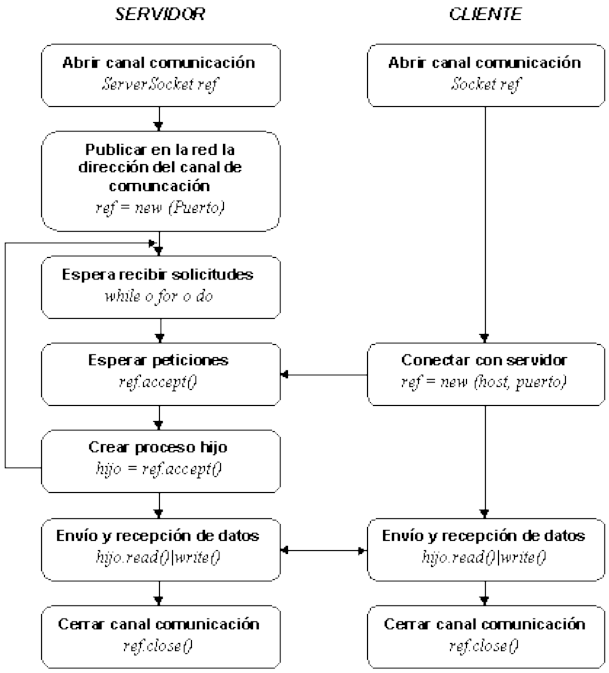

Ilustración 29: Funcionamiento de una conexión con sockets

4.2 Cliente FIRMA

El código fuente implementado para el Cliente FIRMA está agrupado en los siguientes paquetes:

- **Paquete auxiliares:** En este paquete se encuentran las clases auxiliares y sus métodos utilizados en la implementación de este Proyecto.
- **Paquete documento:** Contiene las clases necesarias para realizar la verificación de los diferentes formatos de ficheros con los que trabaja la aplicación FIRMA, fichero de factura PDF y fichero de firma XSIG.
- **Paquete conexión:** Engloba las clases necesarias para poder realizar con éxito la conexión con el Servidor FIRMA y las referidas a la funcionalidad de correo electrónico.
- **Paquete configuración:** Agrupa las clases empleadas para la implementación de la configuración del Cliente FIRMA.
- **Paquete ventanas:** Se encuentran en este paquete todas las clases que conforman el entorno gráfico de la aplicación FIRMA, como por ejemplo, ventanas, cuadros de diálogos, botones, menús, etc.

A continuación se exponen las clases junto con sus respectivos métodos contenidos en cada uno de los anteriores paquetes.

4.2.1 Paquete auxiliares

Este paquete contiene las dos clases auxiliares utilizadas en la implementación de FIRMA. Estas clases son las detalladas a continuación:

- **Clase Blowfish:** Contiene la implementación en Java del algoritmo de cifrado de bloques simétricos del mismo nombre. Se utiliza para almacenar la contraseña del correo electrónico en el fichero de configuración del Cliente FIRMA codificada.

Debido a la gran cantidad de información que se puede encontrar acerca del código fuente de los algoritmos criptográficos no es necesario estudiar dicho código más detenidamente.

- **Clase LimitadorCaracteres:** Clase empleada para controlar la limitación de algunos campos en los formularios en los que el usuario introduce información.

4.2.2 Paquete documento

En este paquete, se encuentran las clases necesarias para la realización de la verificación de autenticidad e integridad tanto de los ficheros de factura PDF como de los ficheros de firma XSIG. Las clases englobadas en este paquete son las siguientes:

- **Clase Resumenes:** En esta clase está implementada la lógica para el cálculo de resúmenes hash según los algoritmos MD5, SHA-1, SHA-256, SHA-384 y SHA-512. Aunque en principio, la aplicación FIRMA sólo emplea el algoritmo SHA-512 para el cálculo del resumen de la factura en formato PDF.
- **Clase Validación:** En esta clase están implementados los métodos necesarios que permiten verificar la relación de dependencia y autenticidad entre los dos ficheros (factura PDF y firma XSIG) a través del resumen del fichero PDF y de la firma contenida en el fichero XSIG.

  Para ello, en primer lugar, se comprueba que la firma contenida en el fichero XSIG es correcta, es decir, la firma corresponde con los datos contenidos en el fichero, con ello se asegura de que no se han producido alteraciones en el fichero de firma.

  Posteriormente, se comprueba que el hash contenido en el fichero de firma pertenece al fichero de factura PDF asociado a dicha firma, para comprobar si se han producido alteraciones en el fichero de factura PDF.

4.2.3 Paquete conexión

Este paquete contiene las dos clases utilizadas relacionadas con la conexión. Estas clases son las siguientes:

- **Clase Mail:** En esta clase está la implementación de la funcionalidad de la aplicación FIRMA de permitir el envío por correo electrónico de una factura junto con su firma, una vez firmada.
- **Clase Cliente:** Contiene la implementación de la lógica de la parte cliente de la arquitectura Cliente-Servidor. A continuación, se muestra un fragmento del código fuente y su correspondiente explicación:

```
public class Cliente  implements ConstantesConfiguracion{

   static Socket socket = null;
   static DataInputStream EntradaSocket;
   static DataOutputStream SalidaSocket;

   // Atributos de la clase cliente para la conexion
   private String ip;
   private int puerto_xsig;

   // Atributo público para tener el nombre del fichero recibido
   public File ficheroRecibido;


   // Modo indica la forma en la que opera el cliente:
   //  -- TRUE --> Si es para tranformar el .pdf en .xsig
   //  -- FALSE--> Si es para validar el .xsig

   public Cliente(boolean modo, File ficheroAEnviar) {

      try {
        // Cargo la configuración del fichero server.properties
        CargarConfiguracion();

        // 1-Paso: Creo el socket
        socket = new Socket(ip,puerto_xsig);

        //Cuando se cierre el socket se espere 10seg para que al cliente
        // le de tiempo recibir los datos
        socket.setSoLinger(true, 10);

        // 2-Paso: Obtengo los flujos de E/S para la comunicación
        InputStream       bufferEntrada = socket.getInputStream();
        OutputStream      bufferSalida = socket.getOutputStream();
        EntradaSocket = new DataInputStream(bufferEntrada);
        SalidaSocket = new DataOutputStream(bufferSalida);

        // 3-Paso: Defino lo que hacer según protocolo con el servidor
        if (modo){
          // Estoy en modo "TRANSFORMAR"
            String mensaje = "TRANSFORMAR";
            SalidaSocket.writeUTF(mensaje);

             // Procedo al envio del fichero .pdf a transformar
             // Primero envio su nombre
             SalidaSocket.writeUTF(ficheroAEnviar.getName());

             // Envio el fichero
             EnviaFichero(ficheroAEnviar);

             //Recojo el fichero .xsig enviado por el servidor
             // Primero recojo el nombre
             ficheroRecibido = new File (EntradaSocket.readUTF());

             // Recojo el fichero
             RecibeFichero(ficheroRecibido);

        }
        else{
             // Estoy en modo "VALIDAR"
             String mensaje = "VALIDAR";
             SalidaSocket.writeUTF(mensaje);
             // Procedo al envio del fichero a comprobar
             // Primero envio su nombre
```

```
            SalidaSocket.writeUTF(ficheroAEnviar.getName());

            // Envio el fichero
            EnviaFichero(ficheroAEnviar);

            // Recojo la respuesta del servidor
            String linea1 = "";
            String linea2 = "";
            String linea3= "";
            linea1 = EntradaSocket.readUTF();
            linea2 = EntradaSocket.readUTF();
            linea3 = EntradaSocket.readUTF();
            // Si las tres son OK→ contable y formato OK
            // Validar la firma y mostrar resultado
            if (linea1.equals(linea2) && (linea2.equals(linea3))){
                  Validacion validacion = new Validacion();
                  validacion.ValidarFactura(ficheroAEnviar);
             }
             else
             // No se ha superado la validación de formato y contable
                 JOptionPane.showMessageDialog(null,linea1,
                   linea2+ficheroAEnviar,Integer.parseInt(linea3));
            }

        // 4-Paso: Cierro la comunicación
        System.out.println("Cerrando conexión...");
        EntradaSocket.close();
        SalidaSocket.close();
        socket.close();
        System.out.println("Conexión cerrada");
      }

      catch (UnknownHostException uhe) {
           uhe.printStackTrace();
          JOptionPane.showMessageDialog(null,"Servidor FIRMA desconocido no
se puede establecer conexión.\n" +
                                "Revise la configuración en Opciones o
póngase en contacto con el Administrador",
                                "Error Servidor FIRMA", 0);

       }
       catch (IOException ioe) {
            ioe.printStackTrace();
            JOptionPane.showMessageDialog(null,"No se ha podido realizar la
conexión con el servidor FIRMA configurado.\n" +
                          "Puede que en estos momentos el servidor no esté
activo",
                          "Error Servidor FIRMA", 0);
       }
   }

   // Caga la config según el fichero para la conexión con el servidor
   private void CargarConfiguracion()

   // Envia el fichero ficheroAEnviar al servidor
   private void EnviaFichero(File ficheroAEnviar) throws IOException

   // Recibe el fichero ficheroRecibido del servidor
   private void RecibeFichero(File ficheroRecibido) throws IOException
}
```

Tabla 15: Fragmento código fuente clase Cliente

En la tabla anterior se puede ver el código de la clase Cliente, la cual en función del modo de operar del Cliente FIRMA define el protocolo de actuación de una forma o de otra, determinado por los dos parámetros que recibe.

En el código anterior hay que destacar el uso del método **`Socket(ip,puerto_xsig)`** el cual sirve para establecer un socket con la máquina cuya IP es el parámetro **`ip`** mediante el puerto indicado por el parámetro **`puertoXSIG`**.

Posteriormente se establecen los correspondientes flujos de entrada/salida en función del socket y se define el protocolo de comunicación con el Servidor, es decir, establece el orden de la comunicación en función de la petición solicitada al Servidor FIRMA. Por ejemplo, en el caso de que un Cliente FIRMA realice una petición de transformar factura de formato PDF a Facturae (se realiza siempre al firmar una factura en formato PDF), el cliente enviará al Servidor FIRMA la cadena **"TRANSFORMAR",** que éste último recogerá. A continuación, enviará una cadena con el nombre del fichero que se desea procesar, el servidor igualmente también la recogerá, y procederá a enviar por bloques el fichero al Servidor FIRMA mediante el método **`EnviaFichero(ficheroAEnviar)`**, que este recoge. Posteriormente, el Servidor FIRMA procederá a tramitar dicha petición y una vez finalizada enviará al Cliente FIRMA el nombre del fichero XSIG y el propio fichero XSIG resultante que el cliente mediante los métodos **`EntradaSocket.readUTF()`** y **`RecibeFichero(ficheroRecibido)`** recogerá.

Los métodos mostrados al final de la tabla, son métodos auxiliares que emplea esta clase para distintos motivos y se apoyan en otras clases definidas en los demás paquetes dentro de Cliente FIRMA. Como su nombre es bastante descriptivo acerca de su funcionalidad y alguno ya se ha explicado en el ejemplo anterior, no se van a detallar detenidamente.

4.2.4 Paquete configuración

En este paquete están recogidas las dos clases relacionadas con la configuración del Cliente FIRMA. Estas clases son las siguientes:

- **Clase ConfiguraciónServer:** Esta clase recoge toda la lógica necesaria para el correcto funcionamiento de la aplicación en función de una configuración determinada recogida en el fichero de configuración del Cliente FIRMA **client.properties,** que será cargada a la aplicación.
- **Clase ConstantesConfiguración:** Es una interface que recoge un grupo de constantes necesarias para los valores de la configuración del Cliente FIRMA.

4.2.5 Paquete ventanas

Este paquete contiene todas las clases que conforman el entorno gráfico de la aplicación FIRMA. Se encuentran las siguientes clases:

- **Clase Principal:** En esta clase está recogida toda la lógica de la ventana principal de la aplicación FIRMA, junto con sus botones, menús, eventos, etc.
  Representa el *JFrame*, o ventana principal de la aplicación, ya que las demás ventanas internas de la aplicación, son *JDialog* y dependen directamente de la principal.
- **Clase CuadroDiálogo:** Esta clase representa la ventana de selección de ficheros que se desean abrir.
- **Clase Documento:** Representa la ventana en la cual se muestra al usuario el fichero de factura PDF que ha seleccionado para firmar gracias al uso del API de ICEpdf.
  Mediante la barra de herramientas que presenta, el usuario puede realizar ciertas operaciones con el fichero PDF (imprimir, guardar, realizar zoom).
  A través de un menú, se le da la opción de enviar por correo electrónico el fichero de factura y la firma XSIG, una vez ya haya realizado la firma.
- **Clase Correo:** Representa al formulario que el usuario debe rellenar para enviar por correo electrónico la factura PDF y su firma (fichero XSIG).
- **Clase Configuración:** Esta clase representa el formulario de configuración de la aplicación. En el que los distintos aspectos a configurar aparecen separados por pestañas.

- **Clase Acerca:** Representa a la ventana de ayuda Acerca de FIRMA de la aplicación.

4.3 Servidor FIRMA

El código fuente implementado para el Servidor FIRMA se agrupa en tres paquetes, estos son los siguientes:

- **Paquete configuración:** Agrupa las clases empleadas para la implementación de la configuración del Servidor FIRMA.
- **Paquete documento:** Se encuentran las clases relacionadas con el tratamiento de los distintos tipos de documentos con los que trabaja el Servidor FIRMA, como son por ejemplo, la factura contenida en el fichero PDF y la factura en formato Facturae versión 3.1.
- **Paquete conexión:** Engloba todas las clases necesarias para poder realizar con éxito la conexión con los Clientes FIRMA y la clase principal que representa al Servidor FIRMA.

A continuación se expone las clases junto con sus respectivos métodos contenidos en cada uno de los anteriores paquetes.

4.3.1 Paquete configuración

En este paquete están recogidas las dos clases relacionadas con la configuración del Servidor FIRMA. Estas clases son las siguientes:

- **Clase ConfiguraciónServer:** Esta clase recoge toda la lógica necesaria para el correcto funcionamiento de la aplicación en función de una configuración determinada recogida en el fichero de configuración del Servidor FIRMA **server.properties,** que será cargada a la aplicación.
- **Clase ConstantesConfiguración:** Es una interface que recoge un grupo de constantes necesarias para los valores de la configuración del Servidor FIRMA.

4.3.2 Paquete documento

Este paquete contiene las clases relacionadas con el tratamiento de los distintos tipos de documentos con los que trabaja el Servidor FIRMA, como por ejemplo, la factura contenida en el fichero PDF y la factura en formato Facturae versión 3.1. En él, se encuentran las clases necesarias para la realización de la transformación entre estos dos formatos y su correspondiente verificación.

Las clases englobadas en este paquete son las siguientes:

- **Clase FacturaPDF:** Esta clase representa a la factura en formato PDF. Incluyendo su tratamiento y posterior proceso de transformación al formato Facturae versión 3.1.

  Para ello, en primer lugar, se extrae del fichero PDF la información contenida en la factura. Seguidamente, gracias el empleo del API Facturae se obtiene un objeto Java que representa a la factura en este formato con la información obtenida anteriormente.

  Posteriormente, se añaden al objeto la información de dos datos fundamentales para la realización de este Proyecto: el nombre del fichero PDF que contiene la factura y su correspondiente resumen o *hash*.

  Finalmente, se crea a partir del objeto Java que representa la factura en formato Facturae, un fichero que contiene a la factura en este formato, obteniendo así el fichero XSIG.

  Debido a la inclusión del nombre del fichero PDF y su *hash* en la factura Facturae del fichero XSIG se obtiene una relación de exclusividad y dependencia entre estos dos ficheros, que permitirá, posteriormente, cualquier tipo de comprobación y verificación de la autenticidad e integridad de estos dos ficheros.
- **Clase Resúmenes:** En esta clase está implementada la lógica para el cálculo de resúmenes o *hash* según los algoritmos MD5, SHA-1, SHA-256, SHA-384 y SHA-512. Aunque en principio, la aplicación FIRMA sólo emplea el algoritmo SHA-512 para el cálculo del resumen de la factura en formato PDF, dato que posteriormente, se integra en la factura en formato Facturae versión 3.1 para su posterior firma y así, poder permitir al usuario verificar la autenticidad e integridad de la factura.

- **Clase Validación:** En esta clase están implementados los métodos necesarios que permiten verificar que los ficheros XSIG se ajustan tanto a nivel de formato como contable al formato determinado por Facturae versión 3.1.

4.3.3 Paquete conexión

En este paquete están englobadas las dos clases utilizadas relacionadas con la conexión de la parte servidor de la arquitectura Cliente-Servidor. Estas clases son las siguientes:

- **Clase Servidor:** Representa la lógica principal del Servidor FIRMA. Se apoya para ello en la clase Flujo, descrita a continuación.

  Contiene el *main* de Servidor FIRMA, y siempre está ejecutándose para escuchar y responder las peticiones que le llegan.

  A continuación, se muestra un fragmento del código fuente y su correspondiente explicación:

```
public class Servidor  implements ConstantesConfiguracion {

   public static void main (String args[]) {

      ServerSocket servidor = null;
       int numClientes = 1; // Contador de clientes
       int puertoXSIG = 0;

       // Carga la configuracion del fichero server.properties
       ConfiguracionServer config = new ConfiguracionServer();
       config.cargarConfiguracion();

       puertoXSIG = Integer.parseInt(config.getValor(PORT_XSIG));

       try{
          servidor = new ServerSocket(puertoXSIG);
          System.out.println ("Servidor: "+InetAddress.getLocalHost());
          System.out.println("Puerto: "+puertoXSIG);

       } catch (IOException ioe) {
             System.out.println("Comunicación rechazada."+ioe);
             System.exit(1);
       }

       System.out.println ("Servidor ejecutandose...");

       while (true) {
          try {
             // 1: Creo el socket para la comunicación
             Socket socket = servidor.accept();
```

```
            System.out.println("Cliente número: "+numClientes);

            //2:Flujos E/S para el proceso-hijo para atender al cliente
            Flujo flujo = new Flujo(socket);

            //3:Resuelvo las peticiones del cliente, pero en un hilo a parte
            Thread t = new Thread(flujo);
            t.start();

            // 4: Cierro la comunicación, cerrada en al fin de t.run()
            // Incremento el contador de clientes
            numClientes++;

         } catch(IOException ioe) {
            System.out.println("Error: "+ioe);
         }
      }
   }
}
```

Tabla 16: Fragmento código fuente clase Servidor

En el código anterior hay que destacar el uso del método **`ServerSocket(puertoXSIG)`**, el cual sirve para atender peticiones de conexiones en el puerto indicado por el parámetro **`puertoXSIG`**.

La razón de englobar el código referente a las peticiones y tratamiento de éstas dentro de un bucle infinito es la de asegurar que el servidor siempre va a estar en ejecución para escuchar peticiones. Una vez que le llega una petición se crea el canal de comunicación con el correspondiente Cliente FIRMA mediante el método **`servidor.accept()`**, y será tratada y procesada en un hilo de ejecución nuevo.

- **Clase Flujo:** Es la clase que representa la lógica del flujo de intercambio de información entre los Clientes FIRMA y el Servidor FIRMA.

  Extiende de la clase predefinida en Java *Thread,* debido a que el Servidor FIRMA debe tratar cada intercambio de información con su correspondiente Cliente FIRMA de manera independiente al resto de clientes, es decir, un *thread* o hilo de ejecución para cada uno de ellos, para así poder responder varias peticiones a la vez.

  Define por lo tanto, una especie de protocolo usado en la forma de comunicación entre los Clientes FIRMA y el Servidor FIRMA.

  En la siguiente tabla puede apreciarse un fragmento del código fuente y su explicación.

```
public class Flujo extends Thread {

   Socket socket;
   DataInputStream FlujoEntrada;
   DataOutputStream FlujoSalida;

   // Constructor
   public Flujo (Socket sfd) {

       socket = sfd;
       try {
          InputStream bufferEntrada = sfd.getInputStream();
          OutputStream     bufferSalida = sfd.getOutputStream();
          FlujoEntrada = new DataInputStream(bufferEntrada);
          FlujoSalida = new DataOutputStream(bufferSalida);

       } catch(IOException ioe) {
          System.out.println("IOException(Flujo): "+ioe);
       }
   }

   public void run() {

       try {
          String respuesta       = "";
          String cadena    = FlujoEntrada.readUTF();
          System.out.println("Mensaje recibido desde cliente: "+cadena);

          if (cadena.equals("VALIDAR")){
          // Procedo a "recoger" el fichero enviado por el cliente
             // Primero recojo el nombre
              File ficheroRecibido = new File (FlujoEntrada.readUTF());

              // Recojo el fichero
              RecibeFichero(ficheroRecibido);

              //Realizo validación del fichero y envío la respuesta al cliente
              ValidarXSIG(ficheroRecibido);

              // Para finalizar borro el fichero de firma que he comprobado
              BorrarFichero(ficheroRecibido);
          }
          else{
              if (cadena.equals("TRANSFORMAR")){
              // Procedo a "recoger" el fichero enviado por el cliente
                 // Primero recojo el nombre
                     File ficheroRecibido = new File (FlujoEntrada.readUTF());

                     // Recojo el fichero
                     RecibeFichero(ficheroRecibido);

                     // Realizo la transformación del fichero
                     File ficheroResultado = GenerarXML(ficheroRecibido);

                     // Envio el firchero .xsig
                     // Primero envio su nombre
                     FlujoSalida.writeUTF(ficheroResultado.getName());

                     // Envio el fichero
                     EnviaFichero(ficheroResultado);

                     // Para finalizar borro el fichero .pdf
                     BorrarFichero(ficheroRecibido);
                     BorrarFichero(ficheroResultado);
```

```
                }
                else{
                      respuesta = "Orden desconocida";
                      FlujoSalida.writeUTF(respuesta);
                      System.out.println("Mensaje respondido: "+respuesta);
                    }
                }

        } catch (IOException e1) {
                e1.printStackTrace();
        }

        // Cierro las conexiones
        try {
                System.out.println("Cerrando conexión...");
                FlujoEntrada.close();
                FlujoSalida.close();
                socket.close();
                System.out.println("Conexión cerrada");
        } catch (IOException e) {
                e.printStackTrace();
        }

}

// Envía por bloques de 1KB el fichero ficheroAEnviar al Cliente
private void EnviaFichero(File ficheroAEnviar) throws IOException

// Recibe por bloques de 1KB el fichero ficheroRecibido del Cliente
private void RecibeFichero(File ficheroRecibido) throws IOException

// Metodo para la validacion del fichero xsig
private void ValidarXSIG (File fichero)

// Método para la eliminación de las copias de los ficheros en el servidor
private void BorrarFichero(File fichero)

// Método para la generación del xsig, devuelve un file con el
// fichero resultante a dicha transformación
private File GenerarXML(File fichero)
```

Tabla 17: Fragmento código fuente clase Flujo

En la tabla anterior se puede ver el código de la clase Flujo, la cual tiene un constructor en el que se establecen los diferentes flujos de entrada/salida en función del socket.

El método **`run()`** es el código a ejecutar cada vez que se inicie un nuevo proceso. Define el protocolo de comunicación con los Clientes, es decir, establece el orden de la comunicación en función de la petición realizada al Servidor FIRMA. Por ejemplo, en el caso de que un Cliente FIRMA realice una petición de validar factura XSIG, el cliente enviará al Servidor FIRMA una vez establecida la comunicación la cadena **"VALIDAR",** que éste último recogerá. A continuación enviará una cadena con el nombre del fichero, el servidor igualmente también la recogerá, y procederá a enviar por bloques el fichero al Servidor FIRMA que este recoge

mediante el método **`RecibeFichero(ficheroRecibido)`**. Posteriormente, el Servidor FIRMA procederá a validar dicho fichero gracias al método **`ValidarXSIG(ficheroRecibido)`**, y para finalizar, enviará la respuesta de la validación al Cliente FIRMA correspondiente que éste recogerá.

Los métodos mostrados al final de la tabla, son métodos auxiliares que emplea esta clase para distintos motivos y se apoyan en otras clases definidas en los demás paquetes dentro de Servidor FIRMA. Como su nombre es bastante descriptivo acerca de su funcionalidad no se van a detallar detenidamente.

V. Test y resultados

En este apartado se recogen las diferentes pruebas realizadas a la aplicación FIRMA. Están organizadas en dos grandes bloques:

- En primer lugar, las comprobaciones relacionadas con la configuración de FIRMA, tanto del Cliente como del Servidor, y
- Por último, las pruebas que se han llevado a cabo para comprobar la funcionalidad de la aplicación: firmar y verificar facturas electrónicas.

5.1 Testing de la configuración

Cliente FIRMA:

- **Prueba 1:** Datos Servidor FIRMA.
  - ✓ Se comprueba que se puedan establecer y/o modificar los diferentes parámetros de la configuración relacionados con el Servidor FIRMA, como son la dirección IP y el puerto en el que se ejecuta.

Ilustración 30: Prueba Datos Servidor FIRMA

- **Prueba 2:** Datos correo electrónico.
  - ✓ Se comprueba que se puedan establecer y/o modificar los diferentes parámetros de la configuración correspondientes con los datos de correo electrónico permitiéndose así utilizar este medio de comunicación para la transmisión de las facturas y sus firmas.

Ilustración 31: Prueba Datos correo electrónico

Servidor FIRMA:

- **Prueba 1:** Configurar Servidor FIRMA.
  - ✓ Se comprueba que se pueda modificar el puerto por defecto por el que se ejecuta el Servidor FIRMA.

Ilustración 32: Prueba Configurar Servidor FIRMA

5.2 Testing de la funcionalidad

Cliente FIRMA:

- **Prueba 1:** Menús y botones de la aplicación.
  - ✓ Se comprueba que los diferentes menús y botones que forman parte de la aplicación funcionan correctamente y cumplen con su cometido.

Ilustración 33: Prueba Menús y botones de la aplicación

- **Prueba 2:** Firmar factura PDF.

- ✓ Se comprueba que el Cliente FIRMA permite firmar digitalmente facturas contenidas en ficheros PDF empleando para ello certificados digitales almacenados tarjetas criptográficas.

Ilustración 34: Prueba Firmar factura PDF

- **Prueba 3:** Verificar factura XSIG.
  - ✓ Se comprueba que el Cliente FIRMA permite verificar una factura y su firma.

Ilustración 35: Prueba Verificar factura XSIG (Cliente)

- **Prueba 4:** Visualizar factura PDF.
  - ✓ Se comprueba que la aplicación Cliente FIRMA permite al usuario visualizar la factura PDF. Permitiéndole, además, realizar ciertas operaciones con el documento (guardar, imprimir, realizar zoom,..).

Ilustración 36: Visualizar factura PDF

- **Prueba 5:** Enviar factura y firma.

- ✓ Se comprueba que el Cliente FIRMA, una vez realizada la firma digital de la factura, permite al usuario enviar tanto la factura como su firma por correo electrónico.

Ilustración 37: Prueba Enviar factura y firma

Servidor FIRMA:

- **Prueba 1:** Transformar factura PDF a formato Facturae versión 3.1.
  - ✓ Se comprueba que el Servidor FIRMA es capaz de realizar la transformación de la factura en formato PDF al formato Facturae versión 3.1 obteniéndose así el fichero XSIG correspondiente.

Ilustración 38: Prueba Transformar factura PDF

- **Prueba 2:** Verificar factura XSIG.
  - ✓ Se comprueba que el Servidor FIRMA permite verificar el fichero XSIG recibido tanto a nivel de formato como contable, ajustándose por lo tanto, al formato Facturae versión 3.1.

Ilustración 39: Prueba Verificar fichero XSIG (Servidor)

VI. Futuros trabajos

Como se ha podido apreciar la aplicación FIRMA dota al usuario de una utilidad para la firma electrónica de facturas junto con su correspondiente verificación a partir del esquema usado en las Administraciones Públicas, Facturae versión 3.1.

Por lo tanto, un posible desarrollo futuro sería la adaptación de FIRMA a futuras versiones del esquema Facturae tanto en el formato de la factura en sí como en el de la firma electrónica.

Para desarrollar FIRMA, se ha partido de la presencia de una factura contenida en un fichero PDF, se podría plantear, según la necesidad del usuario final, desarrollar una nueva funcionalidad para que se pudiese ampliar el formato de ficheros contenedores de facturas, como por ejemplo, una imagen (JPEG), un documento de Microsoft Word (DOC), un hoja de cálculo de Microsoft Excel (XLS) o cualquier otro formato que fuese considerado necesario. En este mismo punto, indicar que una posible futura mejora, podría ser que FIRMA soportase otro tipo de diseño de las facturas diferente al contenido en este Proyecto.

También podría ser útil añadir otras funcionalidades que puedan considerarse de interés para el usuario que complementen a la ya existente del envío por correo electrónico de la factura y su firma digital.

Por ello, la mayoría de los trabajos futuros que se pueden desarrollar a partir de FIRMA están asociados a dar soporte a los cambios en el esquema de Facturae, y la implementación de nuevas utilidades o funcionalidades con las que se quieran completar la aplicación en un momento determinado.

VII. Conclusiones

El desarrollo de un estándar para la factura electrónica por parte de las Administraciones Públicas ha facilitado en gran medida la implantación y utilización de este documento en diferentes áreas.

De esta forma, se ha establecido el punto de partida para todas aquellas aplicaciones y futuros desarrollos relacionados con la facturación electrónica que hasta la aparición de Facturae no estaba muy bien definido como causa del empleo de diversos formatos de factura electrónica.

Así, una vez facilitado un estándar o esquema en el que basarse, se ha conseguido extinguir uno de los principales inconvenientes que presentaba en el principio la factura electrónica e impedía su progresiva implantación, la falta de unicidad de criterios para la correcta representación de las facturas. El esquema Facturae además de solucionar este inconveniente permite dotar a la factura electrónica de la capacidad legal con la que dispone la factura tradicional gracias al empleo de la firma digital, con ello, se garantiza la integridad de los datos que contiene y su procedencia.

Debido al auge de forma exponencial que actualmente está sufriendo la Sociedad de la Información y la necesidad de dotar al ciudadano de funcionalidades que anteriormente requerían su presencia física en determinadas oficinas, por citar algunos ejemplos, trámites de la Seguridad Social, bancos, Agencia Tributaria, conllevan a la utilización de métodos que garanticen fidedignamente la identidad de la persona, como los certificados digitales emitidos por autoridades de certificación reconocidas o bien, más comúnmente el DNI electrónico.

Con el desarrollo de la aplicación FIRMA se proporciona al usuario de una aplicación capaz de firmar electrónicamente facturas contenidas en ficheros PDF, mediante el empleo de certificados digitales, que cumple todos los requisitos legales referidos a la facturación electrónica actual.

Gracias a la arquitectura en la que se ha implementado la aplicación FIRMA, permite tener centralizado en el Servidor las posibles mejoras que en el futuro se produzcan, como pueden ser actualizaciones del formato de factura y firma u otros métodos de verificación de las facturas.

Para finalizar, como opinión personal, indicar que la realización de este Proyecto me ha resultado muy interesante. Gracias a ello, he podido investigar sobre las últimas novedades sobre facturación electrónica y el empleo de tarjetas inteligentes, en principal, de una que en un futuro próximo por no decir hoy mismo, será esencial, el DNI electrónico. Por ello, me gustaría tener la oportunidad, en un futuro, de profundizar más sobre estos temas.

Bibliografía

- Agencia Tributaria, *Portal oficial de la Agencia Tributaria.* [online] Disponible en http://www.agenciatributaria.es/
- Centro de Cooperación Interbancaria, *Portal oficial del Centro de Cooperación Interbancaria (CCI).* [online] Disponible en http://www.asociacioncci.es/cci/es/informacion/
- Desarrolloweb.com, *Manual de Java.* [online] Disponible en http://www.desarrolloweb.com/manuales/57/#capitulos
- Fábrica Nacional de Moneda y Timbre Real Casa de la Moneda, *Portal oficial Fábrica Nacional Moneda y Timbre.* [online] Disponible en http://www.fnmt.es/index.php
- Fábrica Nacional de Moneda y Timbre Real Casa de la Moneda, *Portal oficial Proyecto CERES.* [online] Disponible en http://www.cert.fnmt.es/
- Maestros del web, *Foros del Web (Java).* [online] Disponible en http://www.forosdelweb.com/f45/
- Ministerio de Economía y Hacienda y Ministerio de Industria, Turismo y Comercio, *Portal oficial sobre factura electrónica (Facturae).* [online] Disponible en http://www.facturae.es/es-ES/Paginas/principal.aspx
- Ministerio de Industria, Turismo y Comercio, *Portal oficial sobre el Plan Avanza.* [online] Disponible en http://www.planavanza.es/Paginas/Inicio.aspx
- Ministerio de Industria, Turismo y Comercio, *Portal oficial de la Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información.* [online] Disponible en http://www.mityc.es/dgdsi/es-ES/Paginas/index.aspx
- Ministerio del Interior, *Portal oficial sobre el DNI electrónico.* [online] Disponible en http://www.dnielectronico.es/
- Sun Microsystems. *The Java Tutorials.* [online] Disponible en http://java.sun.com/docs/books/tutorial/index.html

- Sun Microsystems. *Java Platform, Standard Edition 6 API Specification.* [online] Disponible en http://java.sun.com/javase/6/docs/api/
- The Eclipse Foundation, *Portal oficial del Proyecto Eclipse.* [online] Disponible en http://www.eclipse.org/
- Wikimedia Foundation, Inc. *Wikipedia, la enciclopedia libre.* [online] Disponible en http://es.wikipedia.org

Anexo A: Planificación del Proyecto

En este apartado se muestra la información referente a la planificación, asignación de recursos y costes de las diferentes tareas llevadas a cabo para la realización del Proyecto. Gracias a esta información es posible conocer una estimación del tiempo total de ejecución del proyecto y la rentabilidad económica del mismo.

La aplicación utilizada para llevar a cabo esta planificación y presupuesto del proyecto ha sido Microsoft Project.

A.1. Diagrama de Gantt

Seguidamente se muestra una tabla que contiene todas las tareas planificadas que se han ido desarrollando para la finalización de este Proyecto junto con el correspondiente diagrama de Gantt del Proyecto:

Número	Tareas	Predecesoras
1	**PROYECTO FIRMA**	
2	**ANTEPROYECTO**	
3	Estudio de antecedentes	
4	Análisis de requisitos	3
5	Documentación	4
6	**DISEÑO**	**2**
7	Diseño de la arquitectura de la aplicación	
8	Diseño gráfico	7
9	Documentación	8
10	**IMPLEMENTACIÓN**	**6**
11	Crear código fuente ejecutable	
12	Documentación	11
13	**PRUEBAS**	**10**
14	Realización de las pruebas	
15	Corrección de fallos	14
16	Documentación	15
17	**OPERACIÓN**	**13**
18	Recopilación de la documentación	

19	Redacción de la documentación (Memoria)	18

Tabla 18: Tareas del Proyecto

	Nombre de tarea	Duración	Comienzo	Fin	P
1	**PROYECTO FIRMA**	**200 días**	**vie 31/07/09**	**jue 06/05/10**	
2	**ANTEPROYECTO**	**35 días**	**vie 31/07/09**	**jue 17/09/09**	
3	Estudio de antecedentes	20 días	vie 31/07/09	jue 27/08/09	
4	Análisis de requisitos	10 días	vie 28/08/09	jue 10/09/09	3
5	Documentación	5 días	vie 11/09/09	jue 17/09/09	4
6	**DISEÑO**	**27 días**	**vie 18/09/09**	**lun 26/10/09**	2
7	Diseño de la arquitectura de la aplicación	7 días	vie 18/09/09	lun 28/09/09	
8	Diseño gráfico	15 días	mar 29/09/09	lun 19/10/09	7
9	Documentación	5 días	mar 20/10/09	lun 26/10/09	8
10	**IMPLEMENTACIÓN**	**85 días**	**mar 27/10/09**	**lun 22/02/10**	6
11	Crear código fuente ejecutable	80 días	mar 27/10/09	lun 15/02/10	
12	Documentación	5 días	mar 16/02/10	lun 22/02/10	1
13	**PRUEBAS**	**23 días**	**mar 23/02/10**	**jue 25/03/10**	1
14	Realización de las pruebas	16 días	mar 23/02/10	mar 16/03/10	
15	Corrección de fallos	2 días	mié 17/03/10	jue 18/03/10	1
16	Documentación	5 días	vie 19/03/10	jue 25/03/10	1
17	**OPERACIÓN**	**30 días**	**vie 26/03/10**	**jue 06/05/10**	1
18	Recopilación de la documentación	2 días	vie 26/03/10	lun 29/03/10	
19	Redacción de la documentación (Memoria)	28 días	mar 30/03/10	jue 06/05/10	1

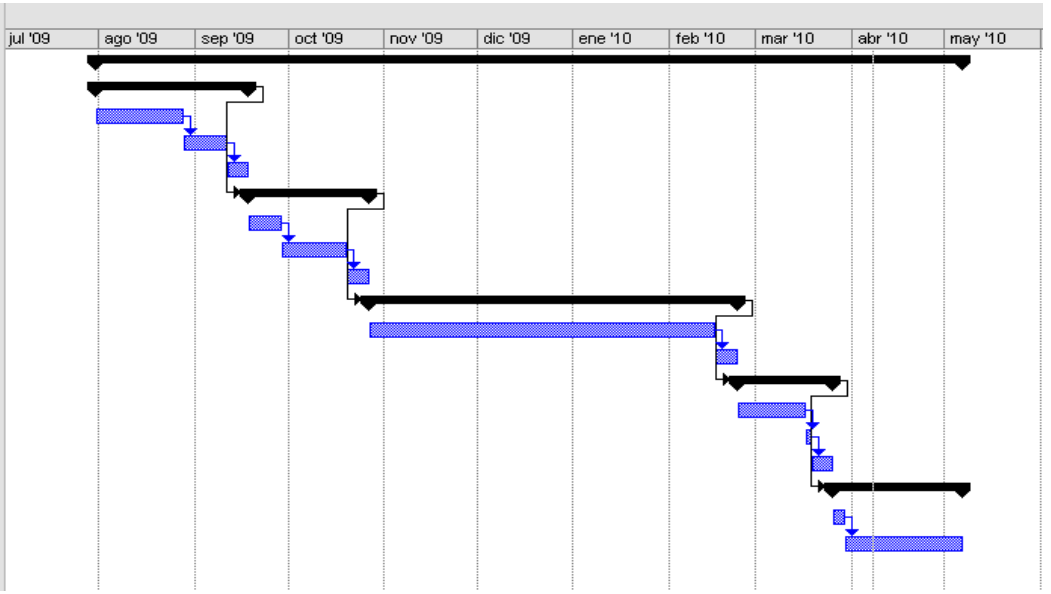

Ilustración 40: Diagrama de Gantt

Del anterior diagrama de Gantt se tiene que precisar la siguiente información relevante del Proyecto:

- **Duración del Proyecto:** 200 días laborables. Entre el 31 de julio de 2009 y el 6 de mayo de 2010, se puede estimar que la duración del Proyecto aproximadamente es de 10 meses tomando, como media, 20 días laborables al mes.
- **Relación de dependencia de tareas con respecto a sus predecesoras.**
- **El coste en recursos humanos de las tareas en función del tiempo empleado en desarrollarlas.** Este coste se puede apreciar tarea por tarea en la siguiente tabla, se toma como unidad de duración el día laborable:

Número	Tareas	Duración	Coste
1	**PROYECTO FIRMA**	**200**	**20.000 €**
2	**ANTEPROYECTO**	**35**	**3.500 €**
3	Estudio de antecedentes	20	2.000 €
4	Análisis de requisitos	10	1.000 €
5	Documentación	5	500 €
6	**DISEÑO**	**25**	**2.700 €**
7	Diseño de la arquitectura de la aplicación	15	700 €
8	Diseño gráfico	20	1.500 €
9	Documentación	5	500 €
10	**IMPLEMENTACIÓN**	**85**	**8.500 €**
11	Crear código fuente ejecutable	80	8.000 €
12	Documentación	5	500 €
13	**PRUEBAS**	**25**	**2.300 €**
14	Realización de las pruebas	20	1.600 €
15	Corrección de fallos	5	200 €
16	Documentación	5	500 €
17	**OPERACIÓN**	**30**	**3.000 €**
18	Recopilación de la documentación	5	200 €
19	Redacción de la documentación (Memoria)	30	2.800 €

Tabla 19: Coste tareas del Proyecto

A.2. Recursos

Los recursos son aquellos “elementos” necesarios para la correcta realización del Proyecto. En este caso, hay que diferenciar entre tres tipos de recursos utilizados:

- **Recursos humanos:** Representan al personal empleado en el desarrollo del Proyecto.
- **Recursos materiales:** Representan a todos los bienes materiales utilizados durante el desarrollo del Proyecto, como por ejemplo, el material informático (PC, periféricos, acceso de Internet), material ofimático, etc.
- **Recursos software:** Incluyen las licencias y el software empleado durante la elaboración del Proyecto.

A.2.1. Recursos humanos

Se ha necesitado durante la elaboración de este Proyecto a un Ingeniero Técnico en Informática de Gestión. Con un calendario laboral de 5 días a la semana de 8 horas (9:00-14:00 horas y 15:00-18:00).

Recursos Humanos	Coste/hora
Ingeniero Técnico en Informática de Gestión:	
Hora normal de desarrollo	12,50 €
Hora extra de desarrollo	14,50 €

Tabla 20: Recursos humanos

A.2.2. Recursos materiales

Para el desarrollo de este Proyecto ha sido necesaria la utilización de los materiales que aparecen recogidos en la siguiente tabla.

Precisar que el coste del ordenador de sobremesa ha sido de 517 euros, pero debido a que tiene una obsolescencia de 36 meses, el coste derivado a la elaboración del Proyecto es el que aparece en la tabla:

Recursos Materiales	Coste
Ordenador de sobremesa: Intel Core 2 Duo 6400 (2,13 GHz) 2 Gb RAM 250 Gb de disco duro Tarjeta gráfica: ATI Radeon HD 3450 Interfaz de red: Ethernet Monitor 19”	**143,61 €** ((517÷36)x10meses)
Cryptokit: Lector de tarjetas inteligentes LTC31 de C3PO Tarjeta criptográfica de la FNMT-RCM	**30,20 €**
DNIe:	**0 €**
Acceso de Internet:	**290 €** (29€/mesx10meses)
TOTAL	**433,61 €**

Tabla 21: Recursos materiales

En cuanto al Documento Nacional de Identidad electrónico (DNIe), como ya se disponía de él con anterioridad al desarrollo de este Proyecto su coste como se ha indicado en la anterior tabla es de 0 euros.

A.2.3. Recursos software

El software utilizado en la realización de este Proyecto es el reflejado en la siguiente tabla. Precisar que el software tiene un ciclo de vida de 3 años:

Recursos Software	Coste
Microsoft Windows XP Professional	129 €
Microsoft Office Professional 2007	512 €
Eclipse	0 €
Java 6 SDK	0 €
SUBTOTAL	**641 €**
TOTAL ASOCIADO AL PROYECTO	**178,06 €** ((641÷36)x10meses)

Tabla 22: Recursos software

A.3. Resumen costes del Proyecto

Con la información contenida en los anteriores apartados se puede obtener el coste económico del desarrollo del Proyecto:

Concepto	Importe
Recursos humanos	20.000 €
Recursos materiales	433,61 €
Recursos software	178,06 €
SUBTOTAL	**20611,67 €**
IVA (16%)	**3297,87 €**
TOTAL	**23909,54 €**

Tabla 23: Costes totales Proyecto

Anexo B: Manual de instalación

Antes de abordar directamente la instalación de la aplicación FIRMA, es necesario disponer en la máquina en la que se vaya a ejecutar de los siguientes requisitos software:

- Máquina virtual Java (JVM): Se ha comprobado la compatibilidad de la aplicación con versiones superiores a la JRE 1.6.0_14. Tanto la máquina virtual Java como el manual de instalación se puede obtener del sitio oficial: http://java.sun.com/javase/downloads/index.jsp
- Drivers del lector de tarjetas inteligentes.
- Módulo criptográfico PKCS#11 para Mozilla.

En estos dos últimos, se dependerá del tipo y fabricante del lector, por ello es necesario seguir su correspondiente manual.

Una vez ya se dispongan de los anteriores requisitos, se procederá a la instalación propiamente dicha de la aplicación FIRMA.

Como se ha indicado en puntos anteriores, la aplicación tiene una arquitectura Cliente-Servidor, por ello, se muestra a continuación el manual de instalación dividido en dos módulos. En primer lugar, el modo de instalación en el servidor y, a continuación, el modo de instalación en el cliente.

Indicar, también, que la aplicación se distribuye en un único fichero .RAR, en el que una vez descomprimido nos encontramos las carpetas “Cliente” y “Servidor”. Cada una de estas carpetas, como bien su nombre indica, se halla contenida la parte de la aplicación correspondiente.

B.1. Manual de instalación del Servidor FIRMA

Para la instalación del Servidor FIRMA únicamente es necesario una vez obtenida dicha carpeta del fichero .RAR de la aplicación, seguir los siguientes pasos:

- **Primer paso, Configurar el Servidor FIRMA:** En caso de ser necesario hay que configurar en el Servidor FIRMA el número de puerto por el que escucha. Solamente hay que acceder al fichero **/Servidor/config/server.properties** y cambiar el valor parámetro **`port_xsig`** por el número de puerto deseado. Indicar que este paso es opcional, en caso de no ser cambiado el número de puerto, el valor por defecto es 50000.

```
# Server config to connection
port_xsig = 50000
```

Tabla 24: Contenido fichero de configuración del Servidor FIRMA (*server.properties)*

- **Segundo paso, Ejecutar el Servidor FIRMA:** Sólo es necesario hacer doble clic en el fichero FIRMA_Servidor.bat. Este fichero .BAT nos permite ejecutar la aplicación desarrollada en Java (.JAR) sin tener la tediosa tarea de ejecutar la aplicación a través de la consola de comandos, lo que facilita enormemente la labor a los usuarios que no disponen de estos conocimientos.

Ilustración 41: Servidor FIRMA ejecutándose

Con estos pasos podemos dar por concluida la instalación de la parte Servidor de la aplicación FIRMA.

B.2. Manual de instalación del Cliente FIRMA

Para la instalación del Cliente FIRMA únicamente es necesario una vez obtenida dicha carpeta del fichero .RAR de la aplicación, seguir los siguientes pasos:

- **Primer paso, Configurar el API Facturae:** Es necesario configurar el API Facturae indicándole la ruta del almacén de certificados de Mozilla de la máquina en la que se va a ejecutar el Cliente de FIRMA y el servidor OCSP que se empleará para la validación de los certificados.

  Para ello, solamente hay que acceder al fichero **/Cliente/config/sign.properties** y cambiar el valor del parámetro **`store.mozilla.profile`** por la ruta del almacén de certificados de Mozilla del PC. Para facilitar su localización se añade el fichero **mozilla.html,** que tras su ejecución con el navegador Mozilla, obtiene dicha ruta.

  Por último, es necesario indicar el servidor OCSP que se empleará para la comprobación y validación de los certificados firmantes.

  Para ello, en el mismo fichero **/Cliente/config/sign.properties** cambiar el valor del parámetro **`ocsp.server`** por la dirección del servidor OCSP que se desee utilizar. Por defecto se empleará el servidor OCSP para validación y comprobación del DNIe.

```
store=mozilla
store.mozilla.profile=C:/Documents and Settings/Administrador/Datos de
programa/Mozilla/Firefox/Profiles/501l2lpa.default

sign.policy=facturae31
sign.xades.schema=1.3.2
locale.language=es
locale.country=ES
lookAndFeel=so

# It will be considered only if lookAndFeel contains "metal"
lookAndFeelTheme=

# OCSP validation
```

```
ocsp.server=http://ocsp.dnie.es/

# Para indicar que no se usa proxy para la conexión
ocsp.proxyused=false

# Para indicar si el proxy es autenticado o no
ocsp.proxyauthenticated=false
ocsp.proxyserver=
ocsp.proxyport=
ocsp.proxyuser=
ocsp.proxypassword=
```

Tabla 25: Contenido del fichero de configuración del API Facturae *(sign.properties)*

- **Segundo paso, Ejecutar el Cliente FIRMA:** Sólo es necesario hacer doble clic en el fichero FIRMA_Cliente.bat. Este fichero .BAT nos permite ejecutar la aplicación desarrollada en Java (.JAR) sin tener la tediosa tarea de ejecutar la aplicación a través de la consola de comandos, lo que facilita la labor al usuario.

Ilustración 42: Cliente FIRMA ejecutándose

- **Tercer paso, Configurar el Cliente FIRMA:** Para disponer plenamente funcional el Cliente de FIRMA es necesario configurar el cliente indicando la dirección IP y puerto en el que el Servidor FIRMA se está ejecutando. Para ello, acceder al menú **Opciones/Configurar.**

Ilustración 43: Captura Opciones/Configurar

Con estos pasos podemos dar por concluida la instalación de la parte Cliente de la aplicación FIRMA.

Anexo C: Manual de usuario

Este manual de usuario está referido a la parte Cliente de FIRMA, ya que la parte Servidor una vez está ejecutándose no necesita de más atención en un funcionamiento normal o sin problemas. Los pasos para la ejecución de la parte Servidor de FIRMA se pueden encontrar en el apartado Manual de instalación del Servidor FIRMA de este mismo documento.

La aplicación FIRMA consta de dos funcionalidades claramente diferenciadas, por un lado tenemos la parte de firma electrónica de la factura y por otro la verificación de la firma de una factura en formato Facturae versión 3.1.

FIRMA ofrece al usuario la posibilidad de enviar, una vez ya firmada la factura, tanto la factura en sí como su firma por correo electrónico al receptor, siempre y cuando los datos de correo estén correctamente configurados en el menú **Opciones/Configurar**.

C.1. Ejecución

Para la ejecución de la aplicación basta con hacer doble clic en el fichero .BAT, FIRMA_Cliente.bat.

Antes de utilizar la aplicación para un uso normal, es recomendable configurar ciertos aspectos de la aplicación que encontramos a continuación.

C.2. Configuración de la aplicación

A continuación se detallarán las opciones para configurar los datos referentes al Servidor FIRMA y las opciones para configurar los datos del correo para poder permitir el envío de las facturas y sus firmas por correo electrónico.

En primer lugar, en caso de que sea necesario cambiar la dirección IP y/o el puerto del Servidor FIRMA de forma posterior a la instalación del Cliente hay que acceder a la ventana **Configuración de FIRMA** pestaña **Datos del servidor** desde el menú **Opciones/Configurar** y cambiar los parámetros necesarios.

Ilustración 44: Configuración de FIRMA (Datos del servidor)

Para la configuración de los datos del correo electrónico con el que permitir el envió de las facturas y sus firmas hay que acceder a la ventana **Configuración de FIRMA** pestaña **Datos del correo** desde el menú **Opciones/Configurar** y cambiar los parámetros necesarios.

Ilustración 45: Configuración de FIRMA (Datos del correo)

A continuación se indican cada uno de estos campos:

- **Nombre o Compañía:** Es el literal que será utilizado por la aplicación para establecer el remitente del correo electrónico enviado.
- **Correo electrónico:** Dirección de correo electrónico empleada para poder realizar el envío.
- **Contraseña:** Contraseña del correo electrónico anterior.
- **Confirmar contraseña:** Repetición de la contraseña del correo electrónico.
- **Servidor SMTP:** Servidor de correo electrónico utilizado para realizar el envío.
- **Puerto SMTP:** Puerto del servidor de correo electrónico empleado para realizar el envío.
- **Autenticación:** Se marcará en el caso de que la dirección de correo empleada requiera contraseña.

C.3. Funcionamiento

A continuación se muestran los pasos necesarios para cada una de las dos funcionalidades que se han indicado que desarrolla la aplicación FIRMA: Firmar la factura y verificar dichas firmas.

C.3.1. Firmar Factura

Una vez ejecutada la aplicación es necesario tener introducido en el lector el DNIe o la tarjeta criptográfica de la FNMT que contenga el certificado con el que se quiere firmar la factura.

Haremos clic en el botón **Firmar** del panel principal de la aplicación o en elemento contenido en el menú **Archivo/Abrir…**.

Ilustración 46: Menú Archivo

Posteriormente, se abrirá un cuadro de diálogo en el que seleccionaremos el fichero con extensión PDF que contiene la factura que deseamos abrir firmar, tal y como se aprecia en la siguiente imagen:

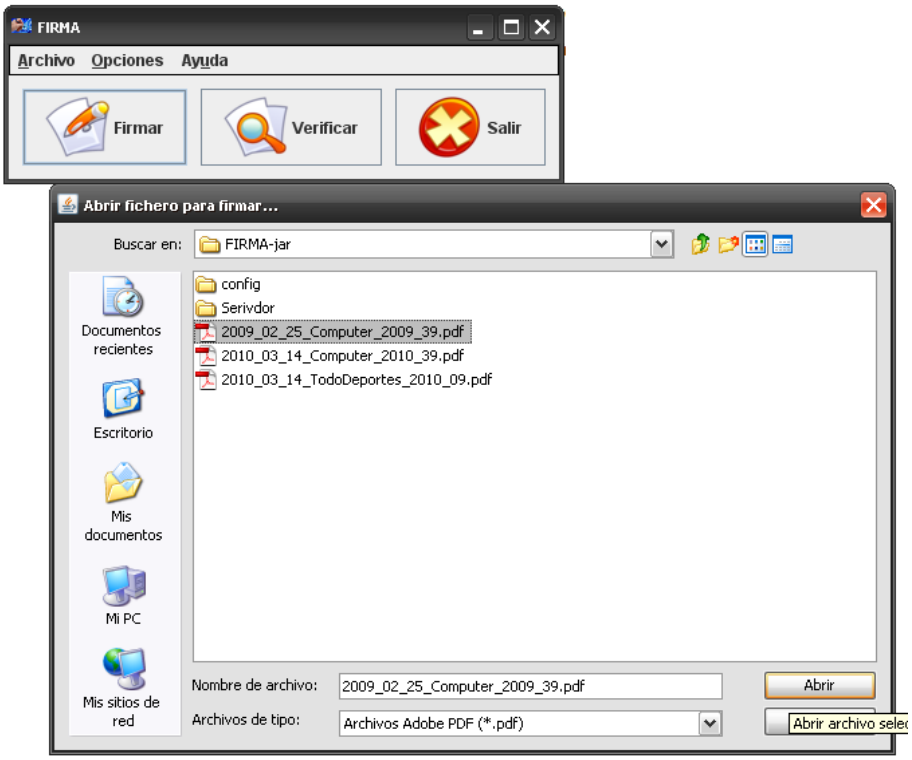

Ilustración 47: Abrir factura fichero PDF

A continuación, se abrirá una ventana mostrándonos el contenido del fichero PDF seleccionado anteriormente, como por ejemplo, la siguiente ilustración:

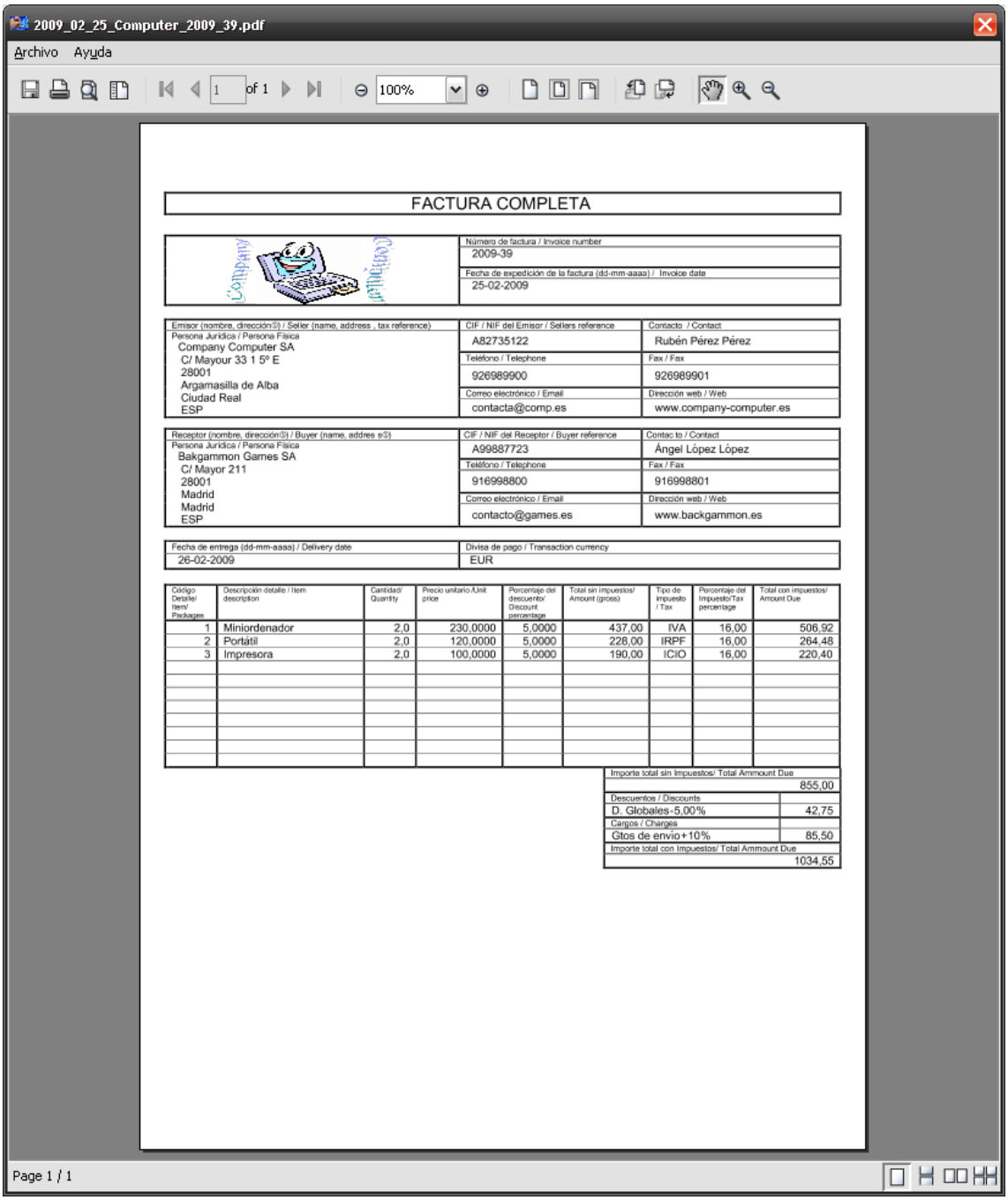

Ilustración 48: Factura fichero PDF

Para proceder a la firma digital de esta factura, tenemos que proceder de la siguiente manera:

1. En el menú archivo **Firmar documento.**

2. Nos aparecerá una ventana solicitándonos la contraseña de la tarjeta contenedora del certificado con el que queremos firmar, o en su defecto, la contraseña de uso del DNIe. Introduciremos la contraseña

Ilustración 49: Solicitud de la contraseña para la firma

3. Seguidamente nos aparecerá una ventana para seleccionar el certificado de la tarjeta con el que queramos firmar la factura, lo seleccionamos y hacemos clic en **Continuar**.

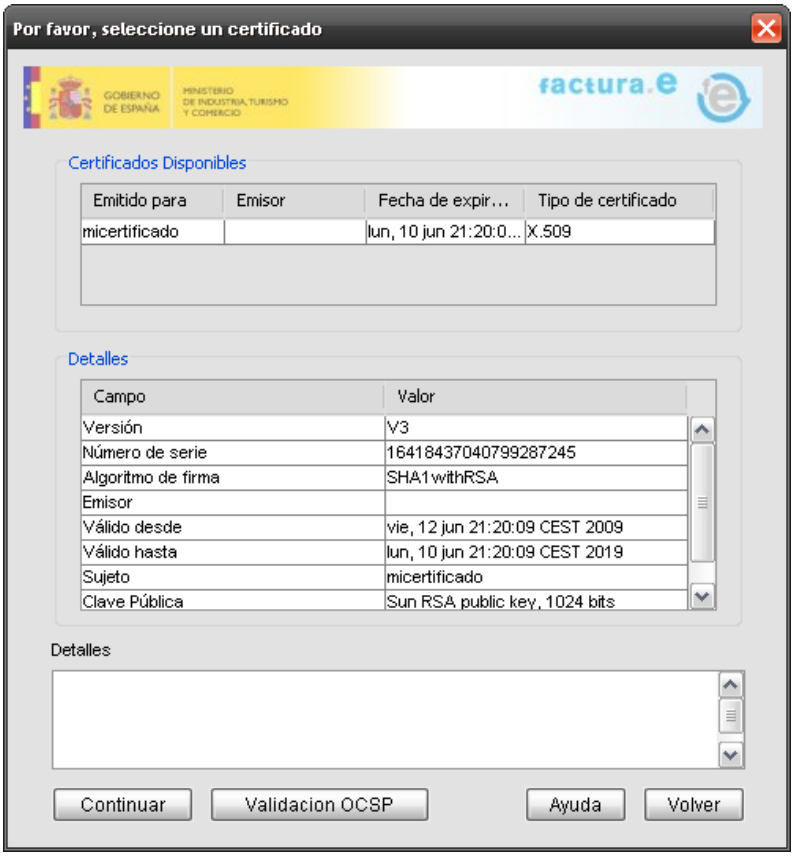

Ilustración 50: Selección de certificado

4. Una vez terminado el proceso de firma nos aparecerá el siguiente aviso:

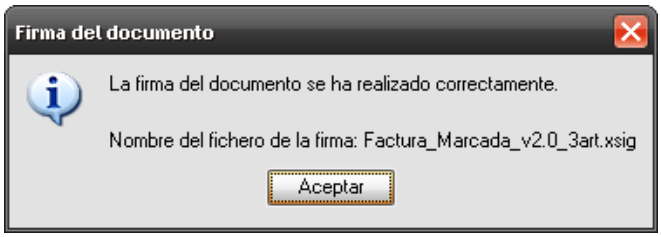

Ilustración 51: Proceso de firma de la factura completado

Mediante el cual se nos informa del nombre con el que se ha generado el fichero de firma en este caso el mismo que el fichero PDF pero con extensión XSIG (Facturae versión 3.1).

Una vez llegado a este punto podemos dar por concluido el proceso de firma de la factura, ahora sólo será necesario enviar al destinatario de la factura el fichero PDF y el fichero XSIG de la firma para que éste pueda comprobar su integridad y autenticidad.

Para realizar el envío por correo electrónico desde la misma aplicación FIRMA bastará seleccionar en el menú **Archivo/Enviar por mail** y rellenar los correspondientes campos del correo electrónico, siempre claro está, estén configuradas correctamente las opciones de correo en el menú **Opciones/Configurar** pestaña **Datos de correo** en la ventana principal y se haya realizado la firma de la factura de forma satisfactoria.

C.3.2. Verificar firma de una factura

Para verificar una factura y su firma que hayamos recibido podemos optar de dos maneras, una haciendo clic en el botón **Verificar** del panel principal de la aplicación o bien, desde el menú **Archivo/Verificar firma**.

Haciendo clic en cualquiera de los dos elementos anteriores, obtenemos un cuadro de diálogo en el que seleccionaremos del fichero XSIG que acompaña a la factura que queremos verificar, como puede apreciarse en la siguiente ilustración:

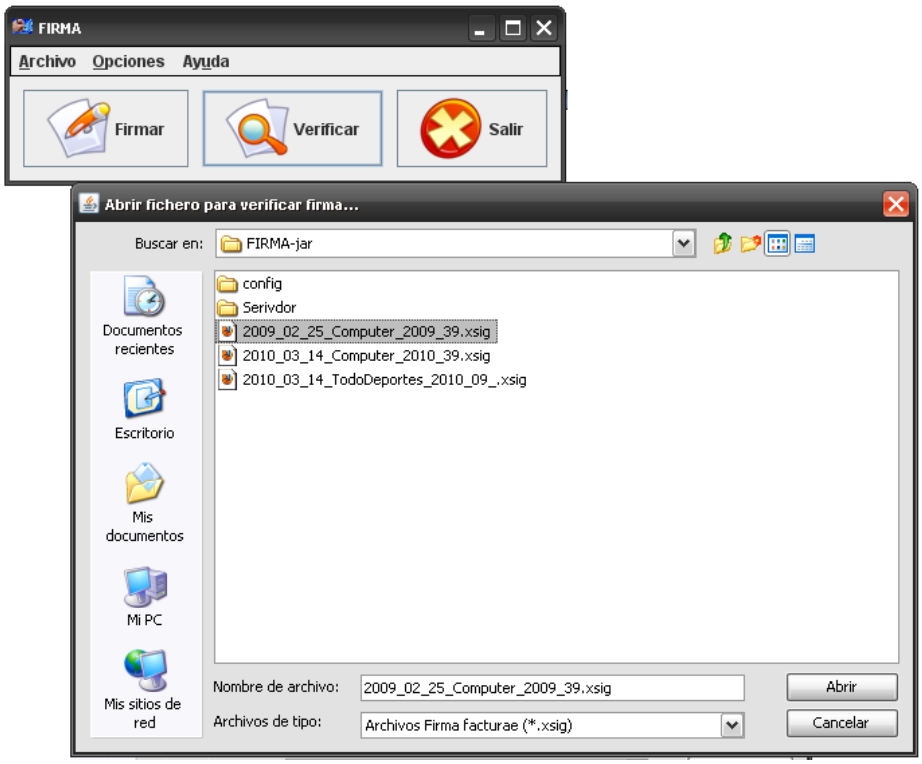

Ilustración 52: Abrir fichero XSIG

Una vez seleccionado el fichero, podemos obtener como resultado los siguientes avisos dependiendo de los siguientes casos:

1. Tanto el fichero de firma como el fichero PDF no han sido modificados después de su firma en origen.

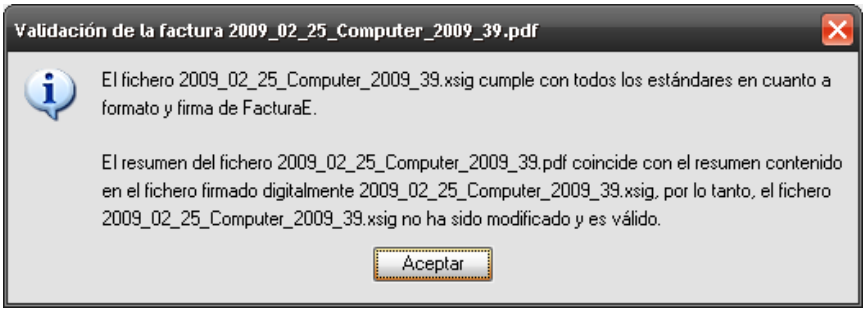

Ilustración 53: Mensaje de fichero de firma correcto

2. Si por cualquier caso, ya sea de forma intencionada o de forma involuntaria se modifica el fichero de firma XSIG, se obtiene el siguiente mensaje:

Ilustración 54: Mensaje de fichero de firma no válido

3. Si se da el caso, en el que el fichero de factura PDF es el que se ha modificado después de la firma se obtiene la advertencia que muestra la siguiente ilustración:

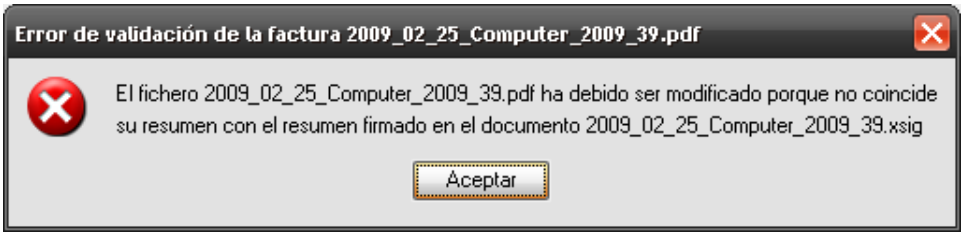

Ilustración 55: Mensaje de fichero de factura no válido

Anexo D: Glosario de términos

A continuación se muestra una tabla con un conjunto de términos y abreviaturas empleados en esta memoria que debido a su importancia merecen ser descritos para su correcta comprensión:

Término	Descripción
AC	**A**utoridad de **C**ertificación.
AEAT	Agencia Tributaria.
API	***A**pplication **P**rogramming **I**nterface.* Conjunto de librerías existentes que proveen una cierta funcionalidad para un lenguaje de programación.
CCI	**C**entro de **C**ooperación **I**nterbancaria. Asociación profesional del colectivo de entidades de depósito (Bancos, Cajas de Ahorros, Cajas Rurales y Cooperativas de Crédito).
CERES	**Cer**tificación **Es**pañola. Proyecto que aglutina a la Entidad Pública de Certificación que permite autentificar y garantizar la confidencialidad de las comunicaciones.
DNIe	**D**ocumento **N**acional de **I**dentidad **e**lectrónico.
Facturae	Esquema adoptado como estándar por las Administraciones Públicas para representar facturas electrónicas.
FNMT-RCM	**F**ábrica **N**acional de **M**oneda y **T**imbre. **R**eal **C**asa de la **M**oneda
Hardware	Designa la parte correspondiente a los circuitos y las máquinas de un sistema.
IP	Número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red del protocolo TCP/IP.
Microescritura	Técnica de seguridad física empleada en billetes o tarjetas, consiste en inscripciones con una altura máxima de caracteres de escritura de aproximadamente 240 µm. La microescritura es tan pequeña que no puede

	descifrarse a simple vista, es necesario para ello el empleo de al menos una lupa o un microscopio.
OCSP	***O**nline **C**ertificate **S**tatus **P**rotocol.* Es un método para determinar el estado de revocación de un certificado digital X.509 usando otros medios que no sean el uso de CRL (Listas de Revocación de Certificados). Este protocolo se describe en el RFC 2560 y está en el registro de estándares de Internet.
PC	***P**ersonal **C**omputer.* Ordenador personal.
PDF	***P**ortable **D**ocument **F**ormat*. Formato de documento portátil, es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.
PIN	***P**ersonal **I**ndentification **N**umber.* Número de identificación personal.
PKCS#11	Interfaz de dispositivo criptográfico. Define un API genérico de acceso a dispositivos criptográficos.
PKI	***P**ublic **K**ey **I**nfrastructure.* Infraestructura de Clave Pública.
Plug-in	Pequeña aplicación que añade funcionalidad a otra aplicación existente; se suele aplicar en forma de parche.
SHA	***S**ecure **H**ash **A**lgorithm.* Familia de algoritmos criptográficos de resumen. Implementaciones concretas son SHA-0, SHA-1 y SHA-2 (de hasta 512 bits).
Smart card o TCI	Tarjeta inteligente o **t**arjeta con **c**ircuito **i**ntegrado.
SMTP	***S**imple **M**ail **T**ransfer **P**rotocol.* Protocolo Simple de Transferencia de Correo, es un protocolo de la capa de aplicación. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos.
Software	Designa la parte correspondiente a los programas de un sistema informático.
XAdES	***X**ML **Ad**vanced **E**lectronic **S**ignatures.* Firma electrónica avanzada XML, es un conjunto de extensiones a las recomendaciones XML DSig haciéndolas adecuadas para la firma electrónica avanzada.
XML	*E**x**tensible **M**arkup **L**anguage.* Lenguaje de marcas extensible, es un metalenguaje extensible de etiquetas desarrollado por el

	World Wide Web Consortium (W3C).
XML DSig	Firma XML (también llamado DSig XML, XML-Sig) es una recomendación del W3C que define una sintaxis XML para la firma digital.
XSIG	Extensión del fichero de firma utilizado por el esquema Facturae versión 3.1.

Tabla 26: Glosario de términos